

INTERNATIONAL STANDARD

ISO/IEC 12207

First edition
1995-08-01

AMENDMENT 1
2002-05-01

Information technology — Software life cycle processes

AMENDMENT 1

Technologies de l'information — Processus du cycle de vie du logiciel
AMENDEMENT 1

Reference number
ISO/IEC 12207:1995/Amd.1:2002(E)



© ISO/IEC 2002

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

© ISO/IEC 2002

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this Amendment may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Amendment 1 to International Standard ISO/IEC 12207:1995 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software engineering*.

Introduction

ISO/IEC 12207 was published on 1 August 1995 and is the first international standard to provide a comprehensive set of life cycle processes, activities and tasks for software that is part of a larger system, stand alone software product, and software services. The standard provides common software process architecture for the acquisition, supply, development, operation and maintenance of software. The standard also provides the necessary supporting processes, activities and tasks, and organizational processes, activities and tasks for managing and improving the processes.

This Amendment provides an interim revision to ISO/IEC 12207 that establishes a co-ordinated set of software process information that can be used for process definition and process assessment and improvement. The Amendment accommodates the requirements of current and developing SC 7 standards and technical reports, notably ISO/IEC 12207 and ISO/IEC/TR 15504, and considers other standards, e.g., ISO/IEC 14598 and ISO/IEC 15939. Experience in using ISO/IEC 12207 as the basis for organizations' software life cycle process and in two-party situations, has resulted in some lessons learned and has provided some valuable inputs to the update process.

During the development of ISO/IEC/TR 15504-2, issues were highlighted in regard to the granularity of the process definition in ISO/IEC 12207, i.e.; it was difficult to derive a process rating component for the purpose of process assessment and improvement. This Amendment resolves this granularity issue and provides process purpose and outcomes to establish a Process Reference Model in accordance with the requirements of ISO/IEC 15504-2. A Process Reference Model provides definitions of processes in a life cycle described in terms of process purpose and outcomes, together with an architecture describing relationships between the processes. A Process Reference Model provides the mechanism whereby externally defined assessment models are related to the assessment framework defined by ISO/IEC 15504.

The current ISO/IEC 12207 process architecture defines the hierarchical relationship among processes, activities and tasks and the invocation rules for the software life cycle processes. Inclusion of a process, an activity, or a task for the Amendment is in accordance and consistent with the existing architecture of ISO/IEC 12207.

Information technology — Software life cycle processes

AMENDMENT 1

Throughout the text:

Change the name of the “Training” process to the “Human Resource” process.

Modify the last sentence of the Foreword to read as follows:

“Annexes A and F form an integral part of this International Standard. Annexes B, C, D, E, G and H are for information only.”

Modify subclause 1.2, paragraph 4, to read as follows:

1.2 Field of Application

This clause does not prevent the use of ISO/IEC 12207 by suppliers or developers of off-the-shelf software.

In subclause 1.4, change “compliance” to read “conformance”.

Add the following text to subclause 1.4:

1.4.1 Conformance to Purposes and Outcomes

Annex F provides an alternative form of conformance useful in situations where implemented processes are intended to achieve the same goals of those described in this standard, but which may not implement the detailed provisions prescribed in the body of this standard. To claim conformance, it shall be demonstrated that, for any process from the set of processes declared by the organization, implementation of the processes results in the realization of the corresponding Purpose and Outcomes provided in Annex F. Any organization shall define the set of processes applicable for it, taking into account the proposed set of processes described in Annex F and its own environmental parameters. Application of the standard allows the creation of additional outcomes.

NOTE In ISO/IEC 12207:1995, the term “compliance” is used in clause 1.4, however, in accordance with ISO/IEC Guide 2, *Standardization and Related Activities — General Vocabulary*, conformance is the appropriate term for this clause. Conformance is the fulfilment by a product, process or service of specified requirements.

Modify subclause 1.5, paragraph 6, to read as follows:

1.5 Limitations

In this International Standard, there are a number of lists for tasks; none of these is presumed to be exhaustive — they are intended as examples unless introduced by a clause containing a “shall” or a “will.”

Add the following reference to clause 2:

ISO/IEC 15504-2, *Software Engineering — Software process assessment — Part 2: Performing an assessment*

Add the following definitions to clause 3:

3.38 Process Purpose: The high level objective of performing the process and the likely outcomes of effective implementation of the process. The implementation of the process should provide tangible benefits to the stakeholders.

3.39 Process Outcome: an observable result of the successful achievement of the process purpose.

NOTE An outcome statement describes one of the following:

- Production of an artefact;
- A significant change in state;
- Meeting of specified constraints, e.g., requirements, goals, etc.

NOTE A list of the principal process outcomes forms part of the description of each process in the reference model.

Add the following subclause to clause 4:

4.2 Relationship of Annex F to the main text of this International Standard

Annex F defines a Process Reference Model (PRM) at a level of abstraction higher than that of the detailed requirements contained in the main text of this International Standard. The PRM is applicable to an organization that is assessing its processes in order to determine the capability of these processes. The Purpose and Outcomes provided in Annex F are a statement of the goals of the performance of each process. This statement of goals permits assessment of the effectiveness of the processes in ways other than simple conformity evaluation. For example, novel process definitions can be evaluated against the statements of Purpose and Outcomes in Annex F rather than against the detailed provisions in the main text of this International Standard.

NOTES

- 1) The term "process reference model" is used with the same meaning as the planned revision of ISO/IEC 15504-2.
- 2) The PRM is intended to be used to develop assessment model(s) for assessing processes using ISO/IEC 15504-2.
- 3) The processes described in Annex F contain extensions, elaborations and some new processes where there is no corresponding development of activities and tasks in ISO/IEC 12207:1995. This will be rectified during the full revision of ISO/IEC 12207:1995. In the meantime, new subclauses 6.9, 7.1.6 and 7.4 to 7.7 provide activities and tasks for the "new" processes of Annex F.

Add the following text to subclause 5.1.1.5:

The acquirer may use Requirements Elicitation sub-process described in Annex F to establish the customer requirements.

Modify subclause 5.1.3.5, sentence 2, as follows:

Subclause 5.1.3.5, "Shall" should be changed to "will"

Add the following text to subclause 5.3.1.2, list item e):

- e) Establish baselines for each configuration item at appropriate times, as determined by the acquirer and the supplier.

Delete sentence 2 of subclause 5.3.4.3.

Delete subclause 5.3.9.5.b.

Delete subclause 5.3.11.4.b.

Add the following text as a second paragraph to the preamble of subclause 6.1:

Execution of this process by an organization results in the establishment of internal documentation standards (such as standards for program management plan and software design document) in a suitable media. The terms used in this process need to be interpreted accordingly for a given media or domain.

Modify line 2 of the preamble of subclause 6.2 as follows:

"Baseline" should be deleted. The resulting sentence should read as follows:

The Configuration Management Process is a process of applying administrative and technical procedures to support the software life cycle to: identify and define software items in a system; control modifications and releases of the items; record and report the status of the items and modification requests; ensure the completeness, consistency, and correctness of the items, and control storage, handling, and delivery of the items.

Replace subclause 6.3.4.1 with the following:

Additional quality management activities can be assured in accordance with the clauses of ISO 9001.

Add the following note to subclause 6.5.2:

NOTE Other means besides testing (such as, analysis, modelling, simulation, etc.) may be employed for validation.

Replace list item e) in subclause 6.6.3.1 with the following:

e) They are ready for the next planned activity.

Add the following references to annex D:

IEEE Std 1517 — 1999, *IEEE Standard for Information Technology — Software Life Cycle Processes — Reuse Processes*

ISO 9000-3, *Quality management and quality assurance standards -- Part 3: Guidelines for the application of ISO 9001:1994 to the development, supply, installation and maintenance of computer software*

ISO 9000: 2000, *Quality management systems — Concepts and vocabulary*

ISO 9001: 2000, *Quality management systems — Requirements*

ISO 9004: 2000, *Quality management systems — Guidance for performance improvement*

ISO/IEC 9126:1991, *Software Product Evaluation — Quality Characteristics and Guidelines for their Use*

ISO 13407:1999, *Ergonomics — Ergonomics of human-system interaction — Human-centred design process for interactive systems*

ISO/IEC 14598:1998, *Software Engineering — Product Evaluation*

ISO/IEC/TR 15504:(all parts), *Information technology — Software process assessment*

ISO/IEC 15504-1, (to be published) *Software Engineering — Software process assessment — Part 1: Concepts and Vocabulary*

ISO/TR 18529, *Ergonomics — Ergonomics of human-system interaction — Human-centred lifecycle process descriptions*

ISO/IEC 15939 *Software Engineering — Software process measurement*

Add the following annexes E, F, G and H:

Annex E **(informative)**

Relationship to ISO 12207:1995

E.1 Relationship of Purpose and Outcomes to ISO/IEC 12207:1995

ISO/IEC 12207:1995 documents the set of software engineering processes that are fundamental to good software engineering and cover best practices. The Processes of the Life Cycle are described in Annex F in terms of the achievement of defined Purposes and Outcomes; these descriptions constitute a reference model, which describes processes that an organization can use to acquire, supply, develop, operate and maintain software. The reference model is also used to provide a common basis for different models and methods for software process assessment, ensuring that the results of the assessments can be reported in a common context. The substantive part of ISO/IEC 12207:1995 sets out the activities and tasks required to implement the high level life cycle processes to achieve desirable capability for acquirers, suppliers, developers, maintainers and operators of systems containing software.

Annex F groups the Purposes and Outcomes into the three life cycle process categories of ISO/IEC 12207:1995, i.e., Organizational, Primary and Supporting. Within each of the process categories are descriptions in terms of a purpose statement, which comprise unique functional objectives when instantiated in a particular environment. The purpose statement includes additional material identifying the outcomes of successful implementation.

Annex F does not define how, or in what order, the elements of the purpose statements are to be achieved. The outcomes will be achieved in an organization through various detailed practices being carried out to produce work products. These performed practices, and the characteristics of the work products produced, are indicators that demonstrate whether the specific purpose is being achieved.

The structure of Annex F and its relationship to the existing International Standard, ISO/IEC 12207:1995, is depicted in Table E-1. For those Purpose and Outcomes that are an "new" to ISO/IEC 12207:1995, descriptions of their activities and/or tasks are provided in new subclauses 6.9, 7.1.6 and 7.4 to 7.7. The activity and task descriptions provided in these new subclauses are in accordance with process structure of ISO/IEC 12207:1995.

E.2 Purpose and Outcomes

The Purpose and Outcomes in Annex F are at the appropriate process, activity or task level to align with the process structure of ISO/IEC 12207. The definition of purpose and outcomes is provided in clause 1.1.2 of this Amendment.

E.3 Process Type

Table E-1 provides a detailed mapping of the content of Annex F to the existing International Standard, ISO/IEC 12207:1995, the source of the information, the structure of the content and the content type. The process structure relationship of Annex F to ISO/IEC 12207 :1995 is defined by process type as follows:

- **Basic** — These processes and sub-processes are identical to the processes and activities of ISO/IEC 12207:1995.
- **New** — These processes and sub-processes are an expansion to the process definition of ISO/IEC 12207:1995.
- **Extended** — These processes and sub-processes are elaborations of the existing processes and activities of ISO/IEC 12207:1995.
- **Component** — These are groupings of existing activities of ISO/IEC 12207:1995.

Table E.1 — Correlation of ISO/IEC 12207:1995 to Annex F

12207	12207 Processes & activities	Annex F Source	Annex F Process Structure	Process Type
5.	<i>Primary life cycle processes</i>			
5.1	Acquisition process	ISO/IEC 12207	Acquisition process	basic
		ISO/IEC/TR 15504-2	Acquisition preparation	component
		ISO/IEC/TR 15504-2	Supplier selection	component
		ISO/IEC/TR 15504-2	Supplier monitoring	component
		ISO/IEC/TR 15504-2	Customer acceptance	component
5.2	Supply process	ISO/IEC 12207	Supply process	basic
5.3	Development process	ISO/IEC 12207	Development process	basic
5.3.1	Process implementation			
		ISO/IEC/TR 15504-2	Requirements elicitation	extended
5.3.2	System requirements analysis	ISO/IEC 12207	System requirements analysis	basic
5.3.3	System architectural design	ISO/IEC 12207	System architectural design	basic
5.3.4	Software requirements analysis	ISO/IEC 12207	Software requirements analysis	basic
5.3.5	Software architectural design	ISO/IEC/TR 15504-2	Software design	component
5.3.6	Software detailed design	ISO/IEC/TR 15504-2	Software design	component
5.3.7	Software coding and testing	ISO/IEC/TR 15504-2	Software construction	component
5.3.8	Software integration	ISO/IEC 12207	Software integration	basic
5.3.9	Software qualification testing	ISO/IEC/TR 15504-2	Software testing	component
5.3.10	System integration	ISO/IEC/TR 15504-2	System integration	component
5.3.11	System qualification testing	ISO/IEC/TR 15504-2	System testing	component
5.3.12	Software installation	ISO/IEC 12207	Software installation	basic
5.3.13	Software acceptance support	ISO/IEC 12207	Supply process	basic
5.4	Operation process	ISO/IEC 12207	Operation process	basic
		ISO/IEC/TR 15504-2	Operational use	extended
		ISO/IEC/TR 15504-2	Customer support	extended
5.5	Maintenance process	ISO/IEC 12207	Maintenance process	basic
6.	<i>Supporting life cycle processes</i>			
6.1	Documentation process	ISO/IEC 12207	Documentation process	basic
6.2	Configuration management process	ISO/IEC 12207	Configuration management process	basic
6.3	Quality assurance process	ISO/IEC 12207	Quality assurance process	basic
6.4	Verification process	ISO/IEC 12207	Verification process	basic
6.5	Validation process	ISO/IEC 12207	Validation process	basic
6.6	Joint review process	ISO/IEC 12207	Joint review process	basic
6.7	Audit process	ISO/IEC 12207	Audit process	basic
6.8	Problem resolution process	ISO/IEC 12207	Problem resolution process	basic
		ISO 13407	Usability process	new
		ISO/IEC 14598	Product evaluation process	extended

Table E.1 — (continued)

12207	12207 Processes & activities	Annex F Source	Annex F Process Structure	Process Type
7.	<i>Organizational life cycle processes</i>			
7.1	Management process	ISO/IEC 12207	Management process	basic
		ISO/IEC/TR 15504-2	Organizational alignment	extended
		ISO/IEC 12207	Organizational management	basic
		ISO/IEC/TR 15504-2	Project management	extended
		ISO/IEC/TR 15504-2	Quality Management	extended
		ISO/IEC/TR 15504-2	Risk Management	extended
		ISO/IEC 15939	Measurement	new
7.2	Infrastructure process	ISO/IEC 12207	Infrastructure process	basic
7.3	Improvement process	ISO/IEC 12207	Improvement process	basic
7.3.1	Process establishment	ISO/IEC/TR 15504-2	Process establishment	component
7.3.2	Process assessment	ISO/IEC/TR 15504-2	Process assessment	component
7.3.3	Process improvement	ISO/IEC/TR 15504-2	Process improvement	component
7.4	Training process	ISO/IEC/TR 15504-2	Human Resource process	new
		ISO/IEC/TR 15504-2	Human resource management	new
		ISO/IEC 12207	Training	basic
			Knowledge management	new
7.5		IEEE 1517	Asset management process	new
7.6		IEEE 1517	Reuse program management process	new
7.7		IEEE 1517	Domain engineering process	new

Annex F (normative)

Purpose and Outcomes

Annex F provides a process reference model that is characterized in terms of process purposes and outcomes, together with an architecture describing the relationships between processes, that describe the expected results from the implementation of this Annex by an organization or a project. The process reference model is applicable to an organization that is assessing processes needed for business success and the subsequent continuous improvement of these processes.

The process model does not represent a particular process implementation approach nor does it prescribe a system/software life cycle model, methodology or technique. Instead the reference model is intended to be tailored by an organization based on its business needs and application domain. The organization's defined process is adopted by the organization's projects in the context of the customer requirements.

The reference model's purpose and outcomes are indicators that demonstrate whether the organization's processes are being achieved. These indicators are useful to process assessors to determine the capability of the organization's implemented process and to provide source material to plan organizational process improvement. The reference model is strongly aligned with ISO/IEC 12207:1995, provides detailed process expectations and includes additional processes determined as essential to enable a reliable and repeatable assessments of software organizations.

NOTE Copyright release: Users may freely reproduce the detailed descriptions of process purpose and outcomes in this annex as part of any Assessment Model based upon the Process Reference Model, or as part of any demonstration of compatibility with the Process Reference Model, so that it can be used for its intended purpose.

F.1 Primary Life Cycle Processes

F.1.1 Acquisition Process

Purpose:

The purpose of the Acquisition Process is to obtain the product and/or service that satisfies the need expressed by the customer. The process begins with the identification of a customer need and ends with the acceptance of the product and/or service needed by the customer.

NOTE Annex H provides an extension of the acquisition process that may be used in lieu of the acquisition process provided in Annex F.

Outcomes:

As a result of successful implementation of the *Acquisition Process* :

- 1) acquisition needs, goals, product and/or service acceptance criteria and acquisition strategies are defined;
- 2) an agreement is developed that clearly expresses the expectation, responsibilities and liabilities of both the customer and the supplier;
- 3) a product and/or service is acquired that satisfies the customer's stated need;
- 4) the acquisition is monitored so that specified constraints such as cost, schedule and quality are met;
- 5) supplier deliverables are accepted;
- 6) any identified open items have a satisfactory conclusion as agreed to by the customer and the supplier.

NOTE Numbering of outcomes is for identification only and does not imply priority or sequence.

The *Acquisition Process* includes purposes and outcomes for the following sub-processes:

- Acquisition Preparation
- Supplier Selection
- Supplier Monitoring
- Customer Acceptance

F.1.1.1 Acquisition preparation

Purpose:

The purpose of *Acquisition preparation* is to establish the needs and goals of the acquisition and to communicate these with the potential suppliers.

Outcomes:

As a result of successful implementation of *Acquisition preparation*:

- 1) the concept or the need for the acquisition, development, or enhancement is established;
- 2) the needed acquisition requirements defining the project needs are defined and validated;
- 3) the customer's known requirements are defined and validated;
- 4) an acquisition strategy is developed; and
- 5) supplier selection criteria are defined.

F.1.1.2 Supplier selection

Purpose:

The purpose of *Supplier selection* is to choose the organization that is to be responsible for the delivery of the requirements of the project.

Outcomes:

As a result of successful implementation of *Supplier selection*:

- 1) the supplier selection criteria are established and used to evaluate potential suppliers;
- 2) the supplier is selected based upon the evaluation of the supplier's proposals, process capabilities, and other factors; and
- 3) an agreement is established and negotiated between the customer and the supplier.

F.1.1.3 Supplier monitoring

Purpose:

The purpose of *Supplier monitoring* is to track and assess performance of the supplier against agreed requirements.

Outcomes:

As a result of successful implementation of *Supplier monitoring*:

- 1) joint activities between the customer and the supplier are performed as needed;
- 2) information on technical progress is exchanged regularly with the supplier;
- 3) performance of the supplier is monitored against the agreed requirements; and
- 4) agreement changes, if needed, are negotiated between the acquirer and the supplier and documented in the agreement.

F.1.1.4 Customer acceptance**Purpose:**

The purpose of *Customer acceptance* is to approve the supplier's deliverable when all acceptance criteria are satisfied.

Outcomes:

As a result of successful implementation of *Customer acceptance*:

- 1) the delivered software product and/or service are evaluated with regard to the agreement
- 2) the customer's acceptance is based on the agreed acceptance criteria; and
- 3) the software product and/or service is accepted by the customer.

F.1.2 Supply Process**Purpose:**

The purpose of the *Supply process* is to provide a product or service to the customer that meets the agreed requirements.

Outcomes:

As a result of successful implementation of the *Supply process*:

- 1) a response to customer's request is produced;
- 2) an agreement is established between the customer and the supplier for developing, maintaining, operating, packaging, delivering, and installing the product and/or service;
- 3) a product and/or service that meets the agreed requirements are developed by the supplier; and
- 4) the product and/or service is delivered to the customer in accordance with the agreed requirements.

F.1.3 Development Process**Purpose:**

The purpose of the *Development Process* is to transform a set of requirements into a software product or software-based system that meets the customer's stated needs. The activities of the Development Process are composed for Systems Developer role and Software Developer role.

Outcomes:

As a result of the successful implementation of the *Development Process* :

- 1) requirements for the development of software are gathered and agreed;
- 2) a software product or software-based system is developed;
- 3) intermediate work products are developed that demonstrate that the end product is based upon the requirements;
- 4) consistency is established between the products of the development process;
- 5) system quality factors are optimized against system requirements, e.g., speed, development cost, usability, etc.;
- 6) evidence (for example, testing evidence) is provided that demonstrates that the end product meets the requirements; and
- 7) the end product is installed in accordance with the agreed requirements.

The *Development Process* includes purposes and outcomes for the following sub-processes:

- Requirements Elicitation
- System Requirements Analysis
- System Architecture Design
- Software Requirements Analysis
- Software Design
- Software Construction (Code and Unit Test)
- Software Integration
- Software Testing
- System Integration
- System Testing
- Software Installation

F.1.3.1 Requirements elicitation

Purpose:

The purpose of *Requirements elicitation* is to gather, process, and track evolving customer needs and requirements throughout the life of the product and/or service so as to establish a requirements baseline that serves as the basis for defining the needed work products. *Requirements elicitation* may be performed by the acquirer or the developer of the system.

Outcomes:

As a result of successful implementation of *Requirements elicitation*:

- 1) continuing communication with the customer is established;
- 2) agreed customer requirements are defined and baselined;

- 3) a change mechanism is established to evaluate and incorporate changes to customer requirements into the baselined requirements based on changing customer needs;
- 4) a mechanism is established for continuous monitoring of customer needs;
- 5) a mechanism is established for ensuring that customers can easily determine the status and disposition of their requests; and
- 6) enhancements arising from changing technology and customer needs are identified and their impact managed.

F.1.3.2 System requirements analysis

Purpose:

The purpose of *System requirements analysis* is to transform the defined stakeholder requirements into a set of desired system technical requirements that will guide the design of the system.

Outcomes:

As a result of successful implementation of *System requirements analysis*:

- 1) a defined set of system functional and non-functional requirements describing the problem to be solved are established;
- 2) the appropriate techniques are performed to optimize the preferred project solution;
- 3) system requirements are analyzed for correctness and testability;
- 4) the impact of the system requirements on the operating environment are understood;
- 5) the requirements are prioritized, approved and updated as needed;
- 6) consistency and traceability is established between the system requirements and the customer's requirements baseline;
- 7) changes to the baseline are evaluated for cost, schedule and technical impact; and
- 8) the system requirements are communicated to all affected parties and baselined.

F.1.3.3 System architectural design

Purpose:

The purpose of *System architectural design* is to identify which system requirements should be allocated to which elements of the system.

Outcomes:

As a result of successful implementation of *System architectural design*:

- 1) a system architecture design is defined that identifies the elements of the system and meets the defined requirements;
- 2) the system's functional and non-functional requirements are addressed;
- 3) the requirements are allocated to the elements of the system;
- 4) internal and external interfaces of each system element are defined;

- 5) verification between the system requirements and the system architecture is performed;
- 6) the requirements allocated to the system elements and their interfaces are traceable to the customer's requirements baseline;
- 7) consistency and traceability between the system requirements and system architecture design is maintained; and
- 8) the system requirements, the system architecture design, and their relationships are baselined and communicated to all affected parties.

F.1.3.4 Software requirements analysis

Purpose:

The purpose of *Software requirements analysis* is to establish the requirements of the software elements of the system.

Outcomes:

As a result of successful implementation of *Software requirements analysis*:

- 1) the requirements allocated to the software elements of the system and their interfaces are defined;
- 2) software requirements are analyzed for correctness and testability;
- 3) the impact of software requirements on the operating environment are understood;
- 4) consistency and traceability are established between the software requirements and system requirements;
- 5) prioritization for implementing the software requirements is defined;
- 6) the software requirements are approved and updated as needed;
- 7) changes to the software requirements are evaluated for cost, schedule and technical impact; and
- 8) the software requirements are baselined and communicated to all affected parties.

F.1.3.5 Software design

Purpose:

The purpose of *Software design* is to provide a design for the software that implements and can be verified against the requirements.

Outcomes:

As a result of successful implementation of *Software design*:

- 1) a software architectural design is developed and baselined that describes the software elements that will implement the software requirements;
- 2) internal and external interfaces of each software elements are defined;
- 3) a detailed design is developed that describes software units that can be built and tested; and
- 4) consistency and traceability are established between software requirements and software design.

F.1.3.6 Software construction**Purpose:**

The purpose of *Software construction* is to produce executable software units that properly reflect the software design.

Outcomes:

As a result of successful implementation of *Software construction*:

- 1) verification criteria are defined for all software units against their requirements;
- 2) software units defined by the design are produced;
- 3) consistency and traceability are established between software requirements and design and software units; and
- 4) verification of the software units against the requirements and the design is accomplished.

F.1.3.7 Software integration**Purpose:**

The purpose of *Software integration* is to combine the software units, producing integrated software items, consistent with the software design, that demonstrate that the functional and non-functional software requirements are satisfied on an equivalent or complete operational platform.

Outcomes:

As a result of successful implementation of *Software integration*:

- 1) an integration strategy is developed for software units consistent with the software design and the prioritized software requirements;
- 2) verification criteria for software items are developed that ensure compliance with the software requirements allocated to the items;
- 3) software items are verified using the defined criteria;
- 4) software items defined by the integration strategy are produced;
- 5) results of integration testing are recorded;
- 6) consistency and traceability are established between software design and software items; and
- 7) a regression strategy is developed and applied for re-verifying software items when a change in software units (including associated requirements, design and code) occur.

F.1.3.8 Software testing**Purpose:**

The purpose of *Software testing* is to confirm that the integrated software product meets its defined requirements.

Outcomes:

As a result of successful implementation of *Software testing*:

- 1) criteria for the integrated software is developed that demonstrates compliance with the software requirements;

- 2) integrated software is verified using the defined criteria;
- 3) test results are recorded; and
- 4) a regression strategy is developed and applied for re-testing the integrated software when a change in software items is made.

F.1.3.9 System integration

Purpose:

The purpose of *System integration* is to integrate the system elements (including software items, hardware items, manual operations, and other systems, as necessary) to produce a complete system that will satisfy the system design and the customers' expectations expressed in the system requirements.

Outcomes:

As a result of successful implementation of *System integration*:

- 1) a strategy is developed to integrate the system according to the priorities of the system requirements;
- 2) criteria is developed to verify compliance with the system requirements allocated to the system elements, including the interfaces between system elements;
- 3) the system integration is verified using the defined criteria;
- 4) a regression strategy is developed and applied for re-testing the system when changes are made;
- 5) consistency and traceability are established between the system design and the integrated system elements; and
- 6) an integrated system, demonstrating compliance with the system design and validation that a complete set of useable deliverable system elements exists, is constructed.

F.1.3.10 System testing

Purpose:

The purpose of *Systems testing* is to ensure that the implementation of each system requirement is tested for compliance and that the system is ready for delivery.

Outcomes:

As a result of successful implementation of *System testing* :

- 1) criteria for the integrated system is developed that demonstrates compliance with system requirements;
- 2) the integrated system is verified using the defined criteria;
- 3) test results are recorded; and
- 4) a regression strategy is developed and applied for re-testing the integrated system should a change be made to existing system elements.

F.1.3.11 Software Installation

Purpose:

The purpose of *Software installation* is to install the software product that meets the agreed requirements in the target environment.

Outcomes:

As a result of successful implementation *Software installation*:

- 1) a software installation strategy is developed;
- 2) criteria for software installation is developed that demonstrates compliance with the software installation requirements;
- 3) the software product is installed in the target environment; and
- 4) assure that the software product is ready for use in its intended environment.

F.1.4 Operation Process**Purpose:**

The purpose of the *Operation Process* is to operate the software product in its intended environment and to provide support to the customers of the software product.

Outcomes:

As a result of the successful implementation of the *Operation Process*:

- 1) conditions for correct operation of the software in its intended environment are identified and evaluated;
- 2) the software is operated in its intended environment; and
- 3) assistance and consultation is provided to the customers of the software product in accordance with the agreement.

The *Operation Process* includes purpose and outcomes for the following sub-processes:

- Operational Use
- Customer Support

F.1.4.1 Operational use**Purpose:**

The purpose of *Operational use* is to ensure the correct and efficient operation of the product for the duration of its intended usage and in its installed environment.

Outcomes:

As a result of successful implementation of *Operational use*:

- 1) operational risks for the product introduction and operation are identified and monitored;
- 2) the product is operated in its intended environment according to requirements; and
- 3) criteria for the operational use are developed that demonstrates compliance with the agreed requirements.

F.1.4.2 Customer support

Purpose:

The purpose of *Customer support* is to establish and maintain an acceptable level of service through assistance and consultation to the customer to support effective use of the product.

Outcomes:

As a result of successful implementation of *Customer support*:

- 1) service needs for customer support are identified and monitored on an ongoing basis;
- 2) customer satisfaction with both the support services being provided and the product itself is evaluated on an ongoing basis;
- 3) operational support is provided by handling customer inquiries and requests and resolving operational problems; and
- 4) customer support needs are met through delivery of appropriate services.

F.1.5 Maintenance Process

Purpose:

The purpose of the *Maintenance process* is to modify a system/software product after delivery to correct faults, improve performance or other attributes, or to adapt to a changed environment.

NOTE The objective is to modify and/or retire existing system/software products while preserving the integrity of organizational operations.

Outcomes:

As a result of successful implementation of the process:

- 1) a maintenance strategy is developed to manage modification, migration and retirement of products according to the release strategy;
- 2) the impact of changes to the existing system on organization, operations or interfaces are identified;
- 3) affected system/software documentation is updated as needed;
- 4) modified products are developed with associated tests that demonstrate that requirements are not compromised;
- 5) product upgrades are migrated to the customer's environment;
- 6) on request, products are retired from use in a controlled manner that minimizes disturbance to the customers; and
- 7) the system/software modification is communicated to all affected parties.

F.2 Supporting Life Cycle Processes

F.2.1 Documentation Process

Purpose:

The purpose of the *Documentation process* is to develop and maintain the recorded software information produced by a process.

Outcomes:

As a result of successful implementation of the *Documentation process*:

- 1) a strategy identifying the documentation to be produced during the life cycle of the software product or service is developed;
- 2) the standards to be applied for the development of the software documentation are identified;
- 3) documentation to be produced by the process or project is identified;
- 4) the content and purpose of all documentation is specified, reviewed and approved;
- 5) documentation is developed and made available in accordance with identified standards; and
- 6) documentation is maintained in accordance with defined criteria.

F.2.2 Configuration Management Process**Purpose:**

The purpose of the *Configuration management process* is to establish and maintain the integrity of all the work products of a process or project and make them available to concerned parties.

Outcomes:

As a result of successful implementation of the *Configuration management process*:

- 1) a configuration management strategy is developed;
- 2) all items generated by the process or project are identified, defined and baselined;
- 3) modifications and releases of the items are controlled;
- 4) modifications and releases are made available to concerned parties;
- 5) the status of the items and modification requests are recorded and reported;
- 6) the completeness and consistency of the items is ensured; and
- 7) storage, handling and delivery of the items are controlled.

F.2.3 Quality Assurance Process**Purpose:**

The purpose of the *Quality assurance process* is to provide assurance that work products and processes comply with predefined provisions and plans.

Outcomes:

As a result of successful implementation of the *Quality assurance process*:

- 1) a strategy for conducting quality assurance is developed;
- 2) evidence of quality assurance is produced and maintained;
- 3) problems and/or non-conformance with agreement requirements are identified and recorded; and
- 4) adherence of products, processes and activities to the applicable standards, procedures and requirements are verified.

F.2.4 Verification Process

Purpose:

The purpose of the *Verification process* is to confirm that each software work product and/or service of a process or project properly reflects the specified requirements.

Outcomes:

As a result of successful implementation of the *Verification process*:

- 1) a verification strategy is developed and implemented;
- 2) criteria for verification of all required software work products is identified;
- 3) required verification activities are performed;
- 4) defects are identified and recorded; and
- 5) results of the verification activities are made available to the customer and other involved parties.

F.2.5 Validation Process

Purpose:

The purpose of the *Validation process* is to confirm that the requirements for a specific intended use of the software work product are fulfilled.

Outcomes:

As a result of successful implementation of the *Validation process*:

- 1) a validation strategy is developed and implemented;
- 2) criteria for validation of all required work products is identified;
- 3) required validation activities are performed;
- 4) problems are identified and recorded;
- 5) evidence is provided that the software work products as developed are suitable for their intended use; and
- 6) results of the validation activities are made available to the customer and other involved parties.

F.2.6 Joint Review Process

Purpose:

The purpose of the *Joint review process* is to maintain a common understanding with the stakeholders of the progress against the objectives of the agreement and what should be done to help ensure development of a product that satisfies the stakeholders. Joint reviews are at both project management and technical levels and are held throughout the life of the project.

Outcomes:

As a result of successful implementation of the *Joint review process*:

- 1) management and technical reviews are held based on the needs of the project;

- 2) the status and products of an activity of a process are evaluated through joint review activities between the stakeholders;
- 3) review results are made known to all affected parties;
- 4) action items resulting from reviews are tracked to closure; and
- 5) problems are identified and recorded.

F.2.7 Audit Process

Purpose:

The purpose of the *Audit process* is to independently determine compliance of selected products and processes with the requirements, plans and agreement, as appropriate.

Outcomes:

As a result of successful implementation of the *Audit process*:

- 1) an audit strategy is developed and implemented;
- 2) compliance of selected software work products and/or services or processes with requirements, plans and agreement is determined according to the audit strategy;
- 3) the conduct of audits by an appropriate independent party are performed; and
- 4) problems detected during an audit are identified and communicated to those responsible for corrective action, and resolution.

F.2.8 Problem Resolution Process

Purpose:

The purpose of the *Problem resolution process* is to ensure that all discovered problems are analyzed and resolved and that trends are recognized.

Outcomes:

As a result of successful implementation of the *Problem resolution process*:

- 1) problem resolution strategy is developed and implemented to ensure that all discovered problems are analyzed and resolved;
- 2) problem reports are prepared upon detection of problems (including non-conformances) in a software product or activity;
- 3) acceptable solutions to recorded problems are identified and acted on; and
- 4) a mechanism is provided for recognizing root cause, implementing defect prevention, and acting on trends in problems identified.

F.2.9 Usability process

Purpose:

The purpose of the *Usability process* is to ensure the consideration of the interests and needs stakeholders in order to enable optimizing support and training, increased productivity and quality of work, improved human working conditions and reducing the chance of user rejection of the system.

Outcomes:

As a result of successful implementation of the *Usability process*:

- 1) the system meets the needs of users and takes account of their human capabilities and skill limitations;
- 2) human factors and ergonomics knowledge and techniques are incorporated in systems design;
- 3) human-centred design activities are identified and performed;
- 4) system design will address possible adverse effects of use on human health, safety and performance; and
- 5) systems will have enhanced user effectiveness, efficiency and satisfaction.

F.2.10 Product Evaluation Process

Purpose:

The purpose of the *Product evaluation process* is to ensure through systematic examination and measurement that a product meets the stated and implied needs of the users of that product.

Outcomes:

As a result of successful implementation of this *Product evaluation process*:

- 1) the requirements for evaluation are established;
- 2) the criteria for product evaluation is identified;
- 3) the methods to be employed for evaluation are defined and the activities needed are identified and performed;
- 4) measures are collected and the results assessed against defined criteria; and
- 5) results of the product evaluation activities are made available to the interested parties.

NOTE Requirements for performing product evaluations are found in ISO/IEC 14598, Software product evaluation. Evaluations may be performed by the Acquirer, the Developer, or a third party Evaluator.

F.3 Organizational Life Cycle Processes

F.3.1 Management Process

Purpose:

The purpose of the *Management process* is to organize, monitor, and control the initiation and performance of any processes to achieve their goals in accord with the business goals of the organization. The *Management process* is established by an organization to ensure the consistent application of practices for use by the organization and the projects. While these practices are inherent to the management of an organization, they are intended to be instantiated for use by each of the organization's projects.

Outcomes:

As a result of successful implementation of the *Management process*:

- 1) the scope of the activity, process to be managed is defined;
- 2) the activities and tasks that must be performed to achieve the purpose of the process are identified;

- 3) the feasibility of achieving process goals with available resources and constraints is evaluated;
- 4) the resources and infrastructure required to perform the identified activities and tasks are established;
- 5) activities are identified and tasks are implemented;
- 6) performance of the defined activities and tasks are monitored;
- 7) work products resulting from the process activities are reviewed and results analyzed and evaluated;
- 8) action is taken to modify the performance of the process when performance deviates from the identified activities and tasks or fails to achieve their goals; and
- 9) successful achievement of the purpose of the process is demonstrated.

The Management Process includes purposes and outcomes for the following sub-processes:

- Organizational Alignment
- Organization Management
- Project Management
- Quality Management
- Risk Management
- Measurement

F.3.1.1 Organizational alignment

Purpose:

The purpose of Organizational alignment is to enable the software processes needed by the organization to provide software products and services, to be consistent with its business goals.

Outcomes:

As a result of the successful implementation of *Organizational alignment*:

- 1) the Organization's business goals are identified
- 2) the process framework is identified and defined that include a set of software processes needed to achieve the business goals of the organization
- 3) a strategy is defined for process definition, implementation and improvement
- 4) support is provided to enable this strategy
- 5) the organization's mission, core values, vision, goals and objectives is made known to all employees;
- 6) individuals in the organization share a common vision, culture, and understanding of the business goals to empower them to function effectively
- 7) everyone in the organization understands their role in achieving the goals of the business and is able to perform that role

F.3.1.2 Organization Management

Purpose:

The purpose *Organization management* is to establish and perform software management practices, during the performance of the processes needed for providing software products and services, that are consistent with the business goals of the organization.

NOTE Although organizational operations in general have a much broader scope than that of software process, software processes are implemented in a business context and to be effective, require an appropriate organizational environment.

Outcomes:

As a result of the successful implementation of *Organization management*:

- 1) the organization will invest in the appropriate management infrastructure;
- 2) the best practices are identified to support the implementation of effective organization and project management; and
- 3) provide a basis for evaluating the achievement of organization business goals based on these management practices.

F.3.1.3 Project management

Purpose:

The purpose of *Project management* is to identify, establish, co-ordinate, and monitor the activities, tasks, and resources necessary for a project to produce a product and/or service, in the context of the project's requirements and constraints.

Outcomes:

As a result of successful implementation of *Project management*:

- 1) the scope of the work for the project is defined;
- 2) the feasibility of achieving the goals of the project with available resources and constraints are evaluated;
- 3) the tasks and resources necessary to complete the work are sized and estimated;
- 4) interfaces between elements in the project, and with other project and organizational units, are identified and monitored;
- 5) plans for the execution of the project are developed and implemented;
- 6) progress of the project is monitored and reported; and
- 7) actions to correct deviations from the plan and to prevent recurrence of problems identified in the project, are taken when project targets are not achieved.

F.3.1.4 Quality management

Purpose:

The purpose of *Quality management* is to achieve customer satisfaction by monitoring the quality of the products and services, at the organizational and project level, to ensure they meet customer requirements.

Outcomes:

As a result of successful implementation of *Quality management*:

- 1) quality goals based on the customer's stated and implicit quality requirements are established;
- 2) an overall strategy is developed to achieve the defined goals;
- 3) a quality management system is established to implement the strategy;
- 4) identified quality control and assurance activities is performed and their performance confirmed;
- 5) actual performance against the quality goals is monitored; and
- 6) appropriate action is taken when quality goals are not achieved.

F.3.1.5 Risk management**Purpose:**

The purpose of *Risk management* is to identify, manage and mitigate the risks continuously, at both the organizational and project level.

Outcomes:

As a result of successful implementation of *Risk management*:

- 1) the scope of the risk management to be performed is determined;
- 2) appropriate risk management strategies are defined and implemented;
- 3) risks to the project are identified in the project's risk management strategy, and as they develop during the conduct of the project;
- 4) the risks are analyzed and the priority in which to apply resources to monitor these risks are determined;
- 5) risk monitoring techniques are selected to determine the change in the risk status and the progress of the monitoring activities; and
- 6) appropriate action is taken to correct or avoid the impact of risk.

F.3.1.6 Measurement**Purpose:**

The purpose of *Measurement* is to collect and analyze data relating to the products developed and processes implemented within the organization and its projects, to support effective management of the processes and to objectively demonstrate the quality of the products.

Outcomes:

As a result of successful implementation of *Measurement*:

- 1) organizational commitment is established and sustained to implement the measurement process;
- 2) the measurement information needs of organizational and management processes are identified;
- 3) an appropriate set of measures, driven by the information needs are identified and/or developed;

- 4) measurement activities are identified and performed;
- 5) the required data is collected, stored, analyzed, and the results interpreted;
- 6) information products are used to support decisions and provide an objective basis for communication; and
- 7) the measurement process and measures are evaluated and communicated to the process owner.

F.3.2 Infrastructure Process

Purpose:

The purpose of the *Infrastructure process* is to maintain a stable and reliable infrastructure that is needed to support the performance of any other process. The infrastructure may include hardware, software, methods, tools, techniques, standards, and facilities for development, operation, or maintenance.

Outcomes:

As a result of successful implementation of the *Infrastructure process*:

- 1) an infrastructure is established that is consistent with and supportive of the applicable process procedures, standards, tools and techniques;
- 2) the infrastructure will meet all requirements for functionality, performance, safety, security, availability, space, equipment, cost, time and data integrity.

F.3.3 Improvement Process

Purpose:

The purpose of the *Improvement process* is to establish, assess, measure, control, and improve a software life cycle process.

Outcomes:

As a result of successful implementation of the *Improvement process*:

- 1) a set of organizational process assets are developed and made available;
- 2) the organization's process capability are assessed periodically to determine the extent to which process implementation is effective in achieving the organization's goals; and
- 3) the effectiveness and efficiency of the organization's processes with respect to business goal achievement are improved on an ongoing basis.

The *Improvement Process* contains purpose and outcomes for the following subprocesses:

- Process Establishment
- Process Assessment
- Process Improvement

F.3.3.1 Process establishment

Purpose:

The purpose of *Process establishment* is to establish a suite of organizational processes for all life cycle processes as they apply to its business activities.

Outcomes:

As a result of successful implementation of *Process establishment*:

- 1) a defined and maintained standard set of processes are established, along with an indication of each process's applicability;
- 2) the detailed tasks, activities and associated work products of the standard process are identified, together with expected performance characteristics;
- 3) a strategy for tailoring the standard process for the product or service is developed in accordance with the needs of the project; and
- 4) information and data related to the use of the standard process for specific projects exist and are maintained.

F.3.3.2 Process assessment**Purpose:**

The purpose of *Process assessment* is to determine the extent to which the organization's standard processes contribute to the achievement of its business goals and to help the organization focus on the need for continuous process improvement.

Outcomes:

As a result of successful implementation of *Process assessment*:

- 1) information and data related to the use of the standard process for specific projects will exist and be maintained;
- 2) the relative strengths and weaknesses of the organization's standard processes are understood; and
- 3) accurate and accessible assessment records are kept and maintained.

F.3.3.3 Process improvement**Purpose:**

The purpose of *Process improvement* is to continually improve the organization's effectiveness and efficiency through the processes used and aligned with the business need.

Outcomes:

As a result of successful implementation of *Process improvement*:

- 1) reviews of the organization's standard processes are carried out at appropriate intervals to ensure their continuing suitability and effectiveness in light of assessment results;
- 2) changes to standard and defined processes are made in a controlled way, with predictable results;
- 3) monitored process improvement activities are implemented in a co-ordinated manner across the organization;
- 4) historical, technical and evaluation data is analyzed and used to improve these processes, to recommend changes in projects, and to determine technology advancement needs; and
- 5) quality cost data is collected, maintained, and used to improve the organization's processes as a monitoring activity, and to serve to establish the cost of prevention and resolution of problems and non-conformity in products and services.

F.3.4 Human Resource Process

Purpose:

The purpose of the *Human resource process* is to provide the organization adequate human resources and maintaining their competencies, consistent with business needs.

Outcomes:

As a result of the successful implementation of the *Human resource process*:

- 1) the roles and skills required for the operations of the organization and the project are identified through timely review of the organizational and project requirements;
- 2) human resources are provided to the organization and the project;
- 3) a set of common training needs across the organization based on organizational and project inputs are identified and provided; and
- 4) the intellectual assets of the organization are made available (or "collected") and exploited through an established mechanism.

Human resource process includes purpose and outcomes for the following subprocesses:

- Human Resource Management
- Training
- Knowledge Management

F.3.4.1 Human Resource Management

Purpose:

The purpose of the *Human resource management process* is to provide the organization and projects with individuals who possess skills and knowledge to perform their roles effectively and to work together as a cohesive group.

Outcomes:

As a result of successful implementation of the *Human resource management*:

- 1) individuals with the required skills and competencies are identified and recruited;
- 2) effective interaction between individuals and groups are supported;
- 3) the work force have the skills to share information and co-ordinate their activities efficiently; and
- 4) objective criteria are defined against which group and individual performance is monitored to provide performance feedback and to enhance performance.

F.3.4.2 Training

Purpose:

The purpose of *Training* is to provide the organization and project with individuals who possess the needed skills and knowledge to perform their roles effectively.

Outcomes:

As a result of successful implementation of *Training*:

- 1) training is developed or acquired to address the organization and project training needs; and
- 2) training is conducted to ensure that all individuals have the skills required to perform their assignments, using mechanisms such as training strategies and materials.

F.3.4.3 Knowledge Management**Purpose:**

The purpose of *Knowledge management* is to ensure that individual knowledge, information and skills are collected, shared, reused and improved throughout the organization.

Outcomes:

As a result of successful implementation of the *Knowledge management*:

- 1) infrastructure is established and maintained for sharing common and domain information across the organization;
- 2) knowledge is readily available and shared throughout the organization; and
- 3) the organization will select an appropriate knowledge management strategy.

F.3.5 Asset Management Process**Purpose:**

The purpose of the *Asset management process* is to manage the life of reusable assets from conception to retirement.

Outcomes:

As a result of successful implementation of the *Asset management process*:

- 1) an asset management strategy is documented;
- 2) an asset classification scheme is established;
- 3) a criteria for asset acceptance, certification and retirement is defined;
- 4) an asset storage and retrieval mechanism is operated;
- 5) the use of assets are recorded;
- 6) changes to the assets are controlled, and
- 7) users of assets are notified of problems detected, modifications made, new versions created and deletion of assets from the storage and retrieval mechanism.

F.3.6 Reuse Program Management Process

Purpose:

The purpose of the *Reuse program management process* is to plan, establish, manage, control, and monitor an organization's reuse program and to systematically exploit reuse opportunities.

Outcomes:

As a result of successful implementation of the *Reuse program management process*:

- 1) define the organization's reuse strategy including its purpose, scope, goals and objectives;
- 2) identify the domains in which to investigate reuse opportunities or in which it intends to practice reuse;
- 3) assess the organization's systematic reuse capability;
- 4) assess each domain to determine its reuse potential;
- 5) evaluate reuse proposals to ensure the reuse product is suitable for the proposed application;
- 6) implement the reuse strategy in the organization;
- 7) establish feedback, communication, and notification mechanisms that operate between reuse program administrators, asset managers, domain engineers, developers, operators, and maintainers; and
- 8) monitor and evaluate the reuse program.

F.3.7 Domain Engineering Process

Purpose:

The purpose of the *Domain engineering process* is to develop and maintain domain models, domain architectures and assets for the domain.

Outcomes:

As a result of successful implementation of the *Domain engineering process*:

- 1) the representation forms for the domain models and the domain architectures are selected;
- 2) the boundaries of the domain and its relationships to other domains are established;
- 3) a domain model that captures the essential common and different features, capabilities, concepts, and functions in the domain are developed;
- 4) a domain architecture describing the family of systems within the domain are developed;
- 5) assets belonging to the domain are specified;
- 6) assets belonging to the domain are acquired or developed and maintained throughout their life cycles; and
- 7) the domain models and architectures are maintained throughout their life cycles.

Annex G (informative)

ISO/IEC 12207:1995 Process Structure for “New” Processes in Annex F

ISO/IEC 12207:1995 defines the process categories, i.e., Organizational, Primary and Supporting Process, that may be performed during the life cycle of software. Within each process category are processes that are expressed in terms of activities and tasks. The activities within a process provide the structural decomposition of the process and describe the actions that are performed during the execution of the process. The tasks within the software life cycle processes provides the "what" is going to be performed during the implementation of the process.

Annex G provides a description of the activities and tasks for the "new" processes identified in Table E.1. These activities and tasks provide in this annex have been an assigned numbering sequence to correspond to the numbering sequence they would have in the body of ISO/IEC 12207:1995. Furthermore, these activities and tasks are in accordance with the process structure of ISO/IEC 12207:1995.

G.1 Supporting life cycle processes

The following process is added to the supporting life cycle processes:

9) Usability Process

G.1.1 Usability Process Activities and Tasks

6.9 Usability process

The Usability Process contains the activities and tasks of the usability specialist. The process contains the activities which take account of the interests and needs of the individuals and/or groups which will work with or use the output from a system throughout development and operation of the software or system. The usability process ensures the quality in use of the software. Details of human-centred design processes can be found in ISO 13407 *Human-centred design processes for interactive systems*. Details of the specification of quality in use can be found in ISO/IEC 9126-1:1999, *Software product quality - Part 1: Quality model*.

The developer manages the Usability Process at the project level. The Usability specialist integrates usability activities and the results from usability activities with the Development (5.3), Operation (5.4) and Supporting (6.3, 6.4, 6.5) lifecycle processes.

List of activities: This process consists of the following activities:

- 1) Process implementation;
- 2) Human-centred design;
- 3) Human aspects of strategy, introduction and support.

NOTE These activities and the associated tasks may overlap or interact and may be performed iteratively or recursively.

6.9.1 Process implementation. This activity consists of the following tasks:

6.9.1.1 Plan and manage the HCD process. Specify how the human-centred activities fit into the whole system lifecycle process and the enterprise.

6.9.1.2 The developer and the usability specialist will:

- a) Consult stakeholders and users.
- b) Identify and plan user involvement.
- c) Select human-centred methods and techniques.
- d) Ensure a human-centred approach within the project team.
- e) Plan human-centred design activities.
- f) Manage human-centred activities.
- g) Champion a human-centred approach.
- h) Provide support for human-centred design.

6.9.2 Human-centred design. This activity consists of the following tasks:

6.9.2.1 A specification of the stakeholder and organisational requirements is provided. Establish the requirements of the organisation and other interested parties for the system. This task takes full account of the needs, competencies and working environment of each relevant stakeholder in the system.

6.9.2.2 In association with the developer the usability specialist will:

- a) Clarify and document system goals.
- b) Analyse stakeholders and users.
- c) Assess the significance and relevance of the system to each stakeholder group.
- d) Assess risk to stakeholders and users.
- e) Define the use of the system.
- f) Generate the stakeholder and organisational requirements.
- g) Set quality in use objectives.

6.9.2.3 An understanding and specification of the context of use is determined. Identify, clarify and record the characteristics of the stakeholders and users, their tasks and the organisational and physical environment in which the system will operate.

6.9.2.4 The usability specialist will:

- a) Identify and document user's tasks.
- b) Identify and document significant user attributes.
- c) Identify and document organisational environment.
- d) Identify and document technical environment.
- e) Identify and document physical environment.

6.9.2.5 The production of design solutions is created. Create potential design solutions by drawing on established state-of-the-art practice, the experience and knowledge of the participants and the results of the context of use analysis.

6.9.2.6 Assisted by the usability specialist the developer will:

- a) Allocate functions.
- b) Produce composite task model.
- c) Explore system design.
- d) Use existing knowledge to develop design solutions.
- e) Specify system and use.
- f) Develop prototypes.
- g) Develop user training.
- h) Develop user support.

6.9.2.7 Evaluation of designs against requirements is determined. Collect feedback on the developing design. This feedback will be collected from end users and other representative sources.

6.9.2.8 The usability specialist will:

- a) Specify and validate context of evaluation.
- b) Evaluate early prototypes in order to define the requirements for the system.
- c) Evaluate prototypes in order to improve the design.
- d) Evaluate the system in order to check that the stakeholder and organisational requirements have been met.
- e) Evaluate the system in order to check that the required practice has been followed.
- f) Evaluate the system in use in order to ensure that it continues to meet organisational and user needs.

6.9.3 Human aspects of strategy, introduction and support. This activity consists of the following tasks:

6.9.3.1 Ensure HCD content in systems strategy. Establish and maintain a focus on stakeholder and user issues in each part of the organisation which deal with system markets, concept, development and support.

The usability specialist will work with relevant marketing and strategy specialists to:

- a) Represent stakeholders and users.
- b) Collect market intelligence.
- c) Define and plan system strategy.
- d) Collect market feedback.
- e) Analyse trends in users.

6.9.3.2 Introduce and operate the system. Establish the human-system aspects of the support and implementation of the system.

6.9.3.3 The usability specialist will work with relevant rollout, training and support specialists to facilitate:

- a) Management of change.
- b) Determination of impact on organisation and stakeholders and users.
- c) Customisation and local design.
- d) Deliver user training.
- e) Support for users in planned activities.
- f) Conformance to workplace ergonomic legislation.

G.2 Management Process

The measurement activity is added to the management process.

G.2.1 Measurement Activity and Tasks

7.1.6 Measurement. This activity consists of the following tasks:

7.1.6.1 The manager will establish and maintain measurement commitment. Ensure that all resource, personnel, and commitment prerequisites for the measurement process have been satisfied. The results of this task provides a commitment from management to support the measurement process, individuals competent in the area of this International Standard have been identified and assigned responsibilities for the measurement process, and resources are made available for measurement planning and performing the process.

7.1.6.2 The manager will plan the measurement process. Develop a detailed plan to initiate, guide, monitor, and evaluate the data collection, analysis, interpretation, and storage tasks. The results of this task provides in planning information that addresses the specific information needs of the organisational unit is defined and any supporting technologies required have been acquired and deployed.

7.1.6.3 The manager will perform measurement in accordance with the plan. Produce information products and performance measures according to the outputs if the measurement planning tasks. The results of this task ensures that data are collected, data are stored in a form suitable for subsequent retrieval and analysis, information products are produced and communicated to the organisational unit, and performance measures are collected.

7.1.6.4 The manager will evaluate measurement. Evaluate the measures and the measurement activities, and store lessons learned from this evaluation in the "Measurement Experience Base". These task results in measures and the measurement activities are evaluated according to specified criteria and lessons learned from this evaluation are stored in the "Measurement Experience Base."

G.3 Human Resource Process Activities and Tasks

The Training Process in ISO/IEC 12207:1995 is renamed the Human Resource Process.

7.4 Human Resource Process

The Human Resource Process provides the organization and projects with individuals who possess skills and knowledge to perform their roles effectively and to work together as a cohesive group.

List of activities: This process consists of the following activities:

- 1) Process Implementation
- 2) Define Training Requirements

- 3) Recruit Qualified Staff
- 4) Evaluate Staff Performance
- 5) Establish Project Team Requirements
- 6) Knowledge Management

7.4.1 Process Implementation. This activity consists of the following task:

7.4.1.1 A review of the organization and project requirements shall be conducted to establish and make timely provision for acquiring or developing the resources and skills required by the management and technical staff. These needs may be met through training, recruitment or other staff development mechanisms.

7.4.2 Define Training Requirements. This activity consists of the following tasks:

7.4.2.1 The types and levels of training and knowledge needed to satisfy organization and project requirements shall be determined. A training plan, addressing implementation schedules, resource requirements, and training needs should be developed and documented.

7.4.2.2 Training manuals, including presentation materials used in providing training should be developed or acquired.

7.4.2.3 Train personnel to have the knowledge and skills needed to perform their roles.

7.4.3 Recruit Qualified Staff. This activity consists of the following task:

7.4.3.1 Establish a systematic program for recruitment of staff qualified to meet the needs of the organization and projects. Provide opportunities for the career development of existing staff.

7.4.4 Evaluate Staff Performance. This activity consists of the following tasks:

7.4.4.1 Define objective criteria that can be used to evaluate staff performance.

7.4.4.2 Evaluate the performance of the staff in respect of their contributions to the goals of the organization or project.

7.4.4.3 Ensure that feedback is provided to the staff on the results of any evaluations performed.

7.4.4.4 Maintain adequate records of staff performance including information on skills, training completed, and performance evaluations.

7.4.5 Establish Team Requirements. This activity consists of the following tasks:

7.4.5.1 Define the organization's and project's need for project teams. Define team structure and operating rules.

7.4.5.2 Empower teams to perform their role by ensuring the teams have:

- a) an understanding of their role on the project;
- b) a shared vision or sense of common interests on the success of the project;
- c) appropriate mechanisms or facilities for communication and interactions among teams; and
- d) support from appropriate management to accomplish project requirements.

7.4.6 Knowledge Management. This activity consists of the following tasks:

7.4.6.1 The manager will plan the requirements for managing the organization's knowledge assets. The planning will include the definition of the infrastructure and training to support the contributors and the users of the organization's knowledge assets, the classification schema for the assets and the asset criteria.

7.4.6.2 The manager will establish a network of experts within the organization. The network will contain the identification of the organization's experts, a list of their area of expertise and the identification of available information within a classification schema, e.g., knowledge area. The manager will ensure that the network is maintained current.

7.4.6.3 The manager will establish a mechanism to support the exchange of information between the experts and the flow of expert information to the organization's projects. The mechanism will support the organization's access, storage and retrieval requirements.

7.4.6.4 Perform configuration management of assets in accordance with the Configuration Management Process specified in subclause 6.2.

G.4 Asset Management Process Activities and Tasks

7.5 Asset Management Process

Regardless of their overall quality and potential for reuse, assets have little value to an organization unless potential reusers know of their existence and can easily locate and understand them.

This process contains the activities and tasks of the asset manager. Asset Management is the process of applying administrative and technical procedures throughout the life of an asset to identify, define, certify, classify, and baseline the asset; track modifications, migrations, and versions of the asset; record and report the status of the asset; and establish and control storage and handling of the asset, delivery of the asset to its reusers, and retirement of the asset.

List of activities. This process consists of the following activities:

- 1) Process implementation;
- 2) Asset storage and retrieval definition;
- 3) Asset management and control.

7.5.1 Process Implementation. This activity consists of the following tasks:

7.5.1.1 The asset manager will create and document an asset management plan, reusing an applicable asset management plan template, if any exists, to define the resources and procedures for managing assets. The plan should include the following:

- a) Defining the requirements for an asset storage and retrieval mechanism;
- b) Defining the asset storage and retrieval mechanism;
- c) Establishing the asset storage and retrieval mechanism as an integral part of the software life cycle;
- d) Naming the organization(s) responsible for managing and maintaining the asset storage and retrieval mechanism;
- e) Defining asset acceptance, certification, and retirement procedures;
- f) Defining the relationship of the asset manager to other parties such as developers, maintainers, and domain engineers;

- g) Promoting the use of the asset storage and retrieval mechanism;
- h) Defining an asset management communication mechanism;
- i) Defining an asset classification scheme.

7.5.1.2 The asset manager will:

- a) Document this activity in accordance with the Documentation Process specified in subclause 6.1;

Perform configuration management of assets in accordance with the Configuration Management Process specified in subclause 6.2;

- b) Document and resolve problems and non-conformance found in the assets and the Asset Management in accordance with the Problem Resolution Process specified in subclause 6.8;
- c) Conduct reviews of assets in accordance with the Joint Review Process specified in subclause 6.6.

7.5.1.3 The asset management plan will be reviewed in accordance with the Joint Review Process specified in subclause 6.6. Domain engineers and reuse program administrators will be included in the review.

7.5.2 Asset storage and retrieval definition. An asset storage and retrieval mechanism enables reusers to easily and quickly find and understand the assets. This activity consists of the following tasks:

7.5.2.1 The asset manager will implement and maintain an asset storage and retrieval mechanism.

7.5.2.2 The asset manager should develop, document, and maintain a classification scheme to be used in classifying the assets.

7.5.2.3 The asset manager will conduct joint review(s) of the asset storage and retrieval mechanism in accordance with the Joint Review Process specified in subclause 6.6. Reuse program administrators and domain engineers will be included in the review(s).

7.5.3 Asset management and control. For each asset, this activity consists of the following tasks:

7.5.3.1 For each asset submitted to the asset manager, the asset will be evaluated based on the asset acceptance and certification criteria.

7.5.3.2 For each asset accepted, it will be made available for reuse through the asset storage and retrieval mechanism.

7.5.3.3 The asset will be classified in accordance with the reuse classification scheme, if any exists.

7.5.3.4 The asset manager will perform configuration management for the asset using the Configuration Management Process specified in subclause 6.2.

7.5.3.5 The asset manager will keep track of each reuse of the asset and report to the domain engineer information about actual reuses of the asset. Asset reuse information should include the reuser's name, project name, original developer or owner of the asset, cost of reusing the asset, and savings and benefits derived from reusing the asset.

7.5.3.6 The asset manager will forward asset modification requests and problem reports received from asset reusers to the domain engineer for review and correction/modification plans and actions. Actions planned and taken to meet requests or to correct problems will be reported to the asset manager making the request or filing the problem report.

7.5.3.7 The asset manager will monitor and record these asset requests/reports and the subsequent actions taken. Whenever problems with an asset are encountered, they should be recorded and entered into the Problem Resolution Process, as specified in subclause 6.8.

7.5.3.8 The asset manager will notify all asset reusers, and the domain engineer of the problems detected in the asset, modifications made to the asset, new versions of the asset, and deletion of the asset from the asset storage and retrieval mechanism.

7.5.3.9 The asset manager will retire assets from the asset storage and retrieval mechanism according to the asset retirement procedures and criteria.

G.5 Reuse Program Management Process Activities and Tasks

7.6 Reuse Program Management Process

Succeeding with the implementation of systematic reuse at the organization level requires careful planning and proper management. Because business, management, and people challenges often outweigh the technical challenges of implementing reuse, management leadership, commitment, and support, as well as a reuse-positive software culture should be emphasized in a reuse program. All individuals within the scope of the reuse program are expected to cooperate with one another in the establishment of reuse processes and to share reuse expertise and assets with one another.

The Reuse Program Management Process contains the activities and tasks of the reuse program administrator. This process is used to plan, establish, manage, control, and monitor an organization's reuse program.

List of activities: This process consists of the following activities:

- 1) Initiation;
- 2) Domain identification;
- 3) Reuse assessment;
- 4) Planning;
- 5) Execution and control;
- 6) Review and evaluation.

7.6.1 Initiation. This activity consists of the following tasks:

7.6.1.1 The reuse program for an organization will be initiated by establishing the organization's reuse strategy that includes its reuse goals, purposes, objectives, and scope. Elements of the reuse program should address the following:

- a) Reuse sponsor;
- b) Reuse infrastructure (including hardware, software, tools, techniques, standards, metrics, and facilities for practising reuse);
- c) Reuse funding and other resources;
- d) Reuse program support function;
- e) Reuse communication, feedback, and notification mechanisms.

NOTE The reuse program administrator defines the following mechanisms:

- a) Feedback mechanism from each software development project to the domain engineer and the asset manager to communicate the use and impact of software products and assets on each project;
- b) Communication mechanism between the software developer, operator, maintainer, domain engineer, asset manager, and the reuse program administrator to resolve problems, answer questions, and make recommendations concerning software products and assets that each project encounters;

- c) Notification mechanism that makes the developer, maintainer, asset manager, and domain engineer aware of the prevailing trade laws, the licensing properties of software products and assets, the organization's restrictions that protect its proprietary interests, and the agreement that may restrict or exclude the use of specific software products or assets by each software development, maintenance, or domain engineering project;
- d) Mechanism for the domain engineer to obtain the participation and information needed from the appropriate sources to complete domain engineering activities.

7.6.1.2 A reuse sponsor should be named.

7.6.1.3 Reuse program participants will be identified and their roles will be assigned.

7.6.1.4 A reuse steering function will be established to assume the authority and responsibility for the organization's reuse program. Its functions should include the following:

- e) Investigation of the practice of reuse in the organization;
- f) Identification of the areas in the organization where there are potential reuse opportunities;
- g) Assignment of the responsibilities for reuse in the organization;
- h) Redefinition of the organization's incentives, disincentives, and culture to support and encourage reuse.

NOTE Members of the reuse steering function include the reuse sponsor, software development manager, operations manager, software maintenance manager, and a reuse expert.

7.6.1.5 A reuse program support function will be established. The responsibilities of the reuse program support function should include the following:

- a) Participating in the creation and implementation of a reuse program implementation plan;
- b) Identifying, documenting, and conveying to all reuse program participants the reuse strategy;
- c) Promoting the practice of reuse to encourage a reuse-positive software culture;
- d) Seeking out opportunities to practice reuse in current and future software projects;
- e) Establishing and maintaining the reuse infrastructure;
- f) Providing reuse consulting support to software projects that practice reuse.

7.6.2 Domain identification

A domain characterizes a set of systems in terms of common properties that can be organized into a collection of reusable assets that may be used to construct systems in the domain. This activity consists of the following tasks:

7.6.2.1 The reuse program administrator, aided by the appropriate manager, domain engineers, users, and software developers, will identify and document the domains in which to investigate reuse opportunities or in which the organization intends to practice reuse.

7.6.2.2 The reuse program administrator, aided by the appropriate managers, domain engineers, users, and software developers, will evaluate the domains to assure that they accurately reflect the organization's reuse strategy. The results of the evaluation will be documented.

7.6.2.3 The reuse program administrator will conduct joint reviews in accordance with the Joint Review Process specified in subclause 6.6. Software developers, domain engineers, and users will be included in the reviews.

7.6.2.4 As more information about the organization's domains and plans for future software products becomes available or when the domains are analyzed, the domains may be refined and rescoped by the reuse program administrator.

7.6.3 Reuse assessment

The reuse assessment provides the baseline against which the practice of reuse in the organization can be measured. Without this reuse assessment, the benefits which result from the practice of reuse in the organization are difficult to measure. The purposes of this activity are to:

- a) Gain an understanding of the reuse maturity of the organization;
- b) Assess the reuse potential of the target domains of the organization;
- c) Make recommendations on how to proceed with the practice of reuse in the organization;
- d) Motivate and direct incremental improvements in the many areas of the organization's reuse program, including reuse training and infrastructure.

This activity consists of the following tasks:

7.6.3.1 The reuse program administrator will assess the organization's systematic reuse capability. The results of the assessment will be documented and provided to the reuse steering function.

7.6.3.2 The reuse program administrator will assess each domain being considered for reuse to determine the potential for reuse success in the domain. The results of the assessment will be documented and provided to the reuse steering function.

7.6.3.3 The reuse program administrator will make recommendations for refining the organization's reuse strategy and reuse program implementation plan based on the results of the reuse assessments. The recommendations will be documented and provided to the reuse steering function.

7.6.3.4 The reuse program administrator, in conjunction with the appropriate acquirers, suppliers, developers, operators, maintainers, asset managers, and domain engineers, will use the Improvement Process specified in subclause 7.3 to incrementally improve the skills, technology, reuse processes, organizational structure, and metrics that together comprise the reuse infrastructure.

7.6.4 Planning. This activity consists of the following tasks:

7.6.4.1 A reuse program implementation plan will be created, documented, and maintained, reusing an applicable reuse program plan template, if any exists, to define the resources and procedures for implementing a reuse program. The plan will describe the following:

- a) The reuse program activities;
- b) Procedures and schedules for performing these activities;
- c) The parties responsible for performing these activities;
- d) The relationships with other parties, such as software developers or domain engineers;
- e) The resources needed for the reuse program;

7.6.4.2 The plan will be reviewed and evaluated considering the following criteria:

- a) Completeness;
- b) Ability to realize the organization's reuse strategy;
- c) Feasibility of implementing the plan.

The results of the evaluation will be documented. Those evaluating the plan should include members of the reuse steering function.

7.6.4.3 Approval and support for the reuse program implementation plan will be obtained from the reuse steering function, and the appropriate managers.

7.6.4.4 The reuse program administrator will conduct joint review(s) in accordance with the Joint Review Process specified in subclause 6.6. Members of the reuse steering function and the appropriate managers will be included in the reviews.

7.6.5 Execution and control. This activity consists of the following tasks:

7.6.5.1 Activities in the reuse program implementation plan will be executed in accordance with the plan.

7.6.5.2 The reuse program administrator will monitor the progress of the reuse program against the organization's reuse strategy, and make and document any necessary adjustments to the plan to realize the strategy.

7.6.5.3 Problems and nonconformances that occur during the execution of the reuse program implementation plan will be recorded and entered into the Problem Resolution Process, as specified in subclause 6.8.

7.6.5.4 The reuse program administrator will periodically reaffirm management sponsorship, support, and commitment to the reuse program.

7.6.6 Review and evaluation. This activity consists of the following tasks:

7.6.6.1 The reuse program administrator will periodically assess the reuse program for achievement of the organization's reuse strategy, and the continued suitability and effectiveness of the reuse program.

7.6.6.2 The reuse program administrator will provide assessment results and lessons learned to the reuse steering function, and to the appropriate managers.

7.6.6.3 The reuse program administrator will recommend and make changes to the reuse program, expand the reuse program, and improve the reuse program in accordance with the Improvement Process specified in subclause 7.3.

G.6 Domain Engineering Process Activities and Tasks

7.7 Domain Engineering Process

The Domain Engineering Process contains the activities and tasks of the domain engineer. The process covers the development and maintenance of the domain models, domain architecture, and other assets for this domain.

List of activities. This process consists of the following activities:

- 1) Process implementation;
- 2) Domain analysis;
- 3) Domain design;
- 4) Asset provision;
- 5) Asset maintenance.

NOTE 1 Domain engineering is a reuse-based approach to defining the scope (i.e., domain definition), specifying the structure (i.e., domain architecture), and building the assets (e.g., requirements, designs, software code, documentation) for a class of systems, subsystems, or applications. Domain engineering may include the following activities: domain definition, domain analysis, developing the domain architecture, and domain implementation.

NOTE 2 These activities and tasks may overlap or interact and may be performed iteratively or recursively. Also, the Domain Engineering Process may overlap with development and maintenance processes that use assets produced by the Domain Engineering Process.

7.7.1 Process implementation. This activity consists of the following tasks:

7.7.1.1 The domain engineer will create and document a domain engineering plan, reusing an applicable domain engineering plan template, if any exists, to define the resources and procedures for performing domain engineering. The plan should include standards, methods, tools, activities, assignments, and responsibilities for performing domain engineering. To create the domain engineering plan, the domain engineer should consult the literature and/or data resources about the domain and should consult with domain experts, developers, and users of software products within the domain. The domain engineering plan will be executed.

7.7.1.2 The domain engineer will select the representation forms to be used for the domain models and domain architectures, in accordance with the organization's reuse standards, and by consulting domain experts, developers, and users of software products within the domain.

7.7.1.3 The domain engineer will:

- a) Document this process in accordance with the Documentation Process specified in subclause 6.1;
- b) Perform configuration management of the domain engineering outputs in accordance the Configuration Management Process specified in subclause 6.2;
- c) Document and resolve problems and non-conformance found in the assets and Domain Engineering Process in accordance with the Problem Resolution Process specified in subclause 6.8;
- d) Conduct joint reviews in accordance with the Joint Review Process specified in subclause 6.6 and include in the review experts of the domain, and software developers and users of software products within the domain;
- e) Establish procedures for receiving, resolving, and providing feedback to the asset manager whenever problems or change requests occur for assets developed by the domain engineer.

7.7.2 Domain analysis. Domain analysis is the activity that discovers and formally describes the commonalities and variabilities within a domain. The domain engineer captures this information in a set of domain models. This activity consists of the following tasks:

7.7.2.1 The domain engineer will define the boundaries of the domain and the relationships between this domain and other domains.

7.7.2.2 The domain engineer will identify the current and anticipated needs of developers of software products within this domain.

7.7.2.3 The domain engineer will build the domain models using the representation forms selected in the Process Implementation Activity for this process.

7.7.2.4 The domain engineer will construct a vocabulary that provides the terminology to describe the important domain concepts and the relationships among similar or common assets of the domain.

7.7.2.5 The domain engineer will classify and document the domain models.

7.7.2.6 The domain engineer will evaluate the domain models and domain vocabulary in accordance with the provisions of the modelling technique selected and in accordance with the organization's asset acceptance and certification procedures. The results of the evaluation will be documented.

7.7.2.7 The domain engineer will conduct domain analysis joint review(s) in accordance with the Joint Review Process specified in subclause 6.6. Software developers, asset managers, domain experts, and users will be included in the reviews.

7.7.2.8 The domain engineer will submit domain models to the asset manager.

7.7.3 Domain design. The domain design activity defines the domain architecture and specifies the assets that can be used to build software products. The domain architecture is a high-level design in which asset interfaces are formally specified. The domain architecture serves as the framework for reusing assets to construct software products. This activity consists of the following tasks:

7.7.3.1 The domain engineer will create and document the domain architecture, consistent with the domain model and following the organization's standards.

7.7.3.2 The domain architecture will be evaluated in accordance with the provisions of the architecture design technique selected and the organization's asset acceptance and certification procedures. The results of the evaluation will be documented.

7.7.3.3 For each entity selected to be designed for reuse, the domain engineer will develop and document an asset specification.

7.7.3.4 For each asset specified, the specification will be evaluated in accordance with the provisions of subclause 5.3.6.7, and in accordance with the organization's asset acceptance and certification procedures. The results of the evaluation will be documented.

7.7.3.5 The domain engineer will conduct domain design joint review(s) in accordance with the Joint Review Process specified in subclause 6.6. Software developers, domain experts, and asset managers will be included in the reviews.

7.7.3.6 The domain engineer will submit the domain architecture to the asset manager.

7.7.4 Asset provision. The asset provision activity develops or acquires assets that can be used to assemble software products. For each asset developed or acquired, this activity consists of the following tasks:

7.7.4.1 The domain engineer will develop the asset, thus

- a) Executing the Acquisition Process (see 5.1) to cause a contract for the asset to be put in place if the asset is to be acquired; or
- b) Executing the Development Process (see 5.3) if the asset is to be developed internally.

7.7.4.2 The asset will be documented and classified.

7.7.4.3 The domain engineer will evaluate the asset in accordance with the organization's asset acceptance and certification procedures. The results of the evaluation will be documented.

7.7.4.4 The domain engineer will conduct asset joint review(s) in accordance with the Joint Review Process specified in subclause 6.6. Software developers and asset managers will be included in the reviews.

7.7.4.5 The domain engineer will submit the asset to the asset manager.

7.7.5 Asset maintenance. The asset maintenance activity contains the tasks for modifying assets, including domain models and domain architectures. An asset undergoes modification to correct a deficiency in the asset or to adapt the asset to meet a new or revised requirement. The domain engineer will modify the asset, executing Maintenance Process specified in subclause 5.5. In addition, the following reuse-related tasks are added to this Maintenance Process when it is applied to maintain an asset:

7.7.5.1 When analyzing requests for asset modification and choosing implementation options, the domain engineer will consider

- a) Conformance with the domain models and the domain architecture;
- b) Impact on the systems and software products that use the asset;

- c) Impact on future users of the asset;
- d) Impact on the reusability of the asset.

7.7.5.2 The domain engineer will obtain approval for the selected implementation option, schedule, and plans to modify the asset.

7.7.5.3 The domain engineer will notify the asset manager who sent the asset modification request about whether the asset modification was approved and the plans and schedule for those approved modifications. When a modification request is not approved, it will be recorded and entered into the Problem Resolution Process, as specified in subclause 6.8.

7.7.5.4 After approval is obtained, the domain engineer will enter the Domain Engineering Process to implement the modifications to an asset.

7.7.5.5 The domain engineer will send the completed modified asset along with any usage instructions and test assets to the asset manager who sent the asset modification request.

Annex H (informative)

ISO/IEC TR 15504-2, PDAM1, Reference Model Extensions For the ISO/IEC 12207:1995 Acquisition Process

Annex H provides the extension of the process definitions in TR 15504-2 for acquirer processes and addresses the current lack of granularity associated with acquisition processes in TR 15504-2. This annex extends the acquirer process definitions in TR 15504-2 and provides the granularity needed for the purpose of acquirer process assessment and acquirer process improvement. These extended processes provide a solid foundation for assessments of acquirer processes and the ability to better ascertain process risks in the procurement of software systems. The acquisition process definitions provided in this annex are included in this Amendment to form the basis for a Process Reference Model to be used with ISO 15504.

The process purposes and process outcomes provided in this annex may be used instead of the process purposes and process outcomes in F.1.1 Acquisition Process. In addition to the Acquisition Process purpose and outcomes, this annex provides the extension of the process definition in the format of ISO/IEC 12207:1995 activities and tasks.

NOTE Copyright release: Users may freely reproduce the detailed descriptions of process purpose and outcomes in this annex as part of any Assessment Model based upon the Process Reference Model, or as part of any demonstration of compatibility with the Process Reference Model, so that it can be used for its intended purpose.

H.1 Acquisition Process Purpose and Outcomes

The following provides an alternative that may be used in lieu of the Annex F1.1 Acquisition Process purpose and outcomes.

Purpose:

The purpose of the Acquisition Process is to obtain the product and/or service that satisfies the need expressed by the customer. The process begins with the identification of a customer need and ends with the acceptance of the product and/or service needed by the customer.

Outcomes:

As a result of successful implementation of the *Acquisition Process*:

- 1) acquisition needs, goals, acceptance criteria and acquisition strategies will be defined;
- 2) an agreement will be developed that clearly expresses the expectation, responsibilities and liabilities of both the customer and the supplier;
- 3) a product and/or service will be produced that satisfies the customer's stated need;
- 4) the acquisition will be monitored so that specified constraints such as cost, schedule and quality are met; and
- 5) supplier deliverables will be accepted.

The subprocesses belonging to the *Acquisition process* are:

- Acquisition Policy
- Acquisition Strategy

- Benefits Analysis
- Technical Requirements
- Legal and Administrative Requirements
- Financial Requirements
- Project Requirements
- Request for Proposals
- Supplier Qualification
- Proposal Evaluation
- Contract Agreement
- Supplier Monitoring
- Acceptance
- Contract Closure
- Supplier Relationships
- User Relationships
- Financial Management

H.1.1 Acquisition Policy

Purpose:

The purpose of the *Acquisition Policy process* is to establish the common high level goals, basis for acquisition needs and the methods to be deployed in the conduct of an acquisition.

Outcomes:

As a result of successful implementation of the process:

- 1) the need to deploy a common acquisition policy for the organisation will be established;
- 2) the systematic basis of, or preference for, technology, process, methods, vendors, standards and legally enforceable regulations to optimise the acquisition will be established;
- 3) the need to ensure adequate resources for managing the acquisition, including the contractual, technical, financial and project management skills of the acquirer will be established;
- 4) the need to define the standards of quality for deliverables acceptable to the stated and implied needs of the acquirer will be established;
- 5) the need to establish an effective and productive relationship with the supplier and other affected groups will be established.

H.1.2 Acquisition Strategy

Purpose:

The purpose of the *Acquisition Strategy process* is to ensure the products to be acquired will comply with the mission, goals and objectives of the business and to provide the basis for planning all aspects of the acquisition project. This process involves a combination of business infrastructure (budgetary, financial investment), acquisition methods (OTS, customised) and common policies (acquisition strategies, schedule determination).

Outcomes:

As a result of successful implementation of the process:

- 1) a planned program management approach for the acquisition that meets the acquisition policy and user / acquirer business needs will be developed;
- 2) specific goals (financial, contract, project, technical) and objectives for different or alternative approaches will be identified;
- 3) the critical success factors for the acquisition will be identified;
- 4) the various ways in which solutions could meet the acquirers needs and expectations will be identified;
- 5) the business risks, financial, technical and resource implications for differing or alternative approaches or solutions will be identified;
- 6) requirements for product updating will be identified.

H.1.3 Benefits Analysis

Purpose:

The purpose of the *Benefits Analysis process* is to establish the continuing relevance and benefit of the acquisition in meeting the evolving and changing needs of the acquirers' requirements and business needs.

Outcomes:

As a result of successful implementation of the process:

- 1) alignment of benefits of the acquisition to business objectives will be analysed;
- 2) analysis of benefits relative to the costs of the acquisition will be performed.

H.1.4 Technical Requirements

Purpose:

The purpose of the *Technical Requirements process* is to establish the technical requirements of the acquisition. This involves the elicitation of functional and non-functional requirements that consider the deployment lifecycle of the products so as to establish a technical requirement baseline.

Outcomes:

As a result of successful implementation of the process:

- 1) the technical requirements, including environment effect evaluation, safety and security requirements where appropriate, will be defined and developed to match the needs and expectations of the users;

- 2) the current and evolving acquisition needs will be gathered and defined;
- 3) the requirements and potential solutions will be communicated to all affected groups;
- 4) a mechanism will be established to incorporate changed or new requirements into the established baseline;
- 5) a mechanism for identifying and managing the impact of changing technology to the technical requirements will be defined;
- 6) the requirements will include compliance with relevant standards, including environment effect evaluation, safety and security standards where appropriate.

NOTE ISO 9126 may be a useful model to elicit technical requirements

H.1.5 Legal and Administrative Requirements

Purpose:

The purpose of the *Legal and Administrative Requirements process* is to define the awarding aspects — expectations, liabilities, legal and other issues and which comply with national and international laws of contract.

Outcomes:

As a result of successful implementation of the process:

- 1) a contractual approach will be defined which is compliant with relevant national, international and regulatory laws, guidance and policies;
- 2) an agreement (contractual) terms and conditions will be defined to describe how the supplier will meet the needs and expectations;
- 3) acceptance criteria and mechanisms for handling of breaches to the fulfilment of contract will be established;
- 4) the rights of the acquirer to assume, modify or evaluate, directly or indirectly Intellectual Property Rights will be established;
- 5) warranties and service level agreements will be provided for where applicable;
- 6) provision for the suppliers to deliver other requirements (e.g. quality plan, escrow arrangements etc) will be defined;
- 7) Recognised criteria for proprietary, regulatory and other product liabilities issues will be established.

NOTE Final settlement of terms will be determined during the Contract Establishment subprocess

H.1.6 Financial Requirements

Purpose:

The purpose of the *Financial Requirements process* is to specify the requirements to prepare the infrastructure for an effective financial management of the acquisition project.

Outcomes:

As a result of successful implementation of the process:

- 1) financial management, risks and costs to the acquirer will be established ;
- 2) financial terms for costs and payments governing the acquisition will be defined and recorded;

- 3) financial aspects of the contract award process will be traceable to the outcome;
- 4) requests for financing will be used to prepare budgets for project activities subject to authorised budgetary controls;
- 5) requirements for cost reporting with the supplier will be established against agreed cost estimation model(s);
- 6) requirements for payments to be managed in accordance with a defined procedure that interrelates to contract data and achievement from project management will be established;
- 7) prioritisation of requirements will be made to ensure that the acquisition lifecycle solution is aligned with relative importance of requirements.

H.1.7 Project Requirements

Purpose:

The purpose of the *Project Requirements process* is to specify the requirements to ensure the acquisition project are performed with adequate planning, staffing, directing, organising and control over project tasks and activities.

Outcomes:

As a result of successful implementation of the process:

- 1) consistency between financial, technical, contract and project requirements will be established;
- 2) requirements for the organisational, management, controlling, and reporting aspects of a project will be defined;
- 3) requirements for adequate staffing of projects by a competent team (e.g. legal, contractual, technical, project competent resources) with clear responsibilities and goals will be defined;
- 4) the needs for exchanging information between all affected parties will be established;
- 5) requirements for the completion and acceptance of interim work products and release of payments will be established;
- 6) risks associated with project lifecycle and with suppliers will be identified;
- 7) requirements for ownership of interactions and relationships with suppliers will be defined;
- 8) rights for use and distribution of the product by the customer and supplier will be defined; and
- 9) support and maintenance requirements will be established.

H.1.8 Request for Proposals

Purpose:

The purpose of the *Request for Proposals process* is to prepare and issue the necessary acquisition requirements. The documentation will include, but not be limited to, the contract, project, finance and technical requirements to be provided for use in the Call For Proposals (CFP) / Invitation To Tender (ITT).

Outcomes:

As a result of successful implementation of the process:

- 1) rules will be defined for proposal/tender invitation and evaluation which comply with the acquisition policy and strategy;

- 2) the baseline technical and non-technical requirements will be assembled to accompany the CFP / ITT;
- 3) the agreement (contractual) terms of reference and conditions for CFP / ITT will be established;
- 4) the financial terms of reference for costs and payments for CFP / ITT will be defined;
- 5) the project terms of reference for CFP / ITT will be defined;
- 6) the technical terms of reference for CFP / ITT will be defined;
- 7) a CFP / ITT will be prepared and issued in accordance with acquisition policies and which complies with relevant national, international and regulatory laws, requirements, and policies.

H.1.9 Supplier Qualification

Purpose:

The purpose of the *Supplier Qualification Evaluation process* is to evaluate and determine if the potential supplier(s) have the required qualification for entering the proposal / tender evaluation process. In this process, the technical background, quality system, servicing, user support capabilities and etc will be evaluated.

Outcomes:

As a result of successful implementation of the process:

- 1) criteria will be established for qualifying suppliers ;
- 2) supplier capability determination will be performed as necessary;
- 3) the suppliers which possess required qualification will be short-listed for tender solution(s) evaluation;
- 4) any shortfalls will be identified and evaluated;
- 5) corrective action required by the acquirer will be evaluated and performed.

H.1.10 Proposal Evaluation

Purpose:

The purpose of the *Proposal Evaluation process* is to evaluate proposed / tendered solutions, and/or associated Off The Shelf (OTS) products in order to enter into contract / agreement negotiations.

Outcomes:

As a result of successful implementation of the process:

- 1) the proposed / tendered solutions will be evaluated against the CFP / ITT requirements;
- 2) criteria will be established for qualifying Off The Shelf (OTS) products where these are offered as (part of) a proposed / tendered solution;
- 3) OTS products will be evaluated as necessary against a defined plan to determine the degree of fit with the acquirers needs and expectations;
- 4) the supplier(s) of the successful proposed / tendered solution(s) will be invited to enter into contract / agreement negotiation.

H.1.11 Contract Agreement

Purpose:

The purpose of the *Contract Agreement process* is to negotiate and approve a contract / agreement that clearly and unambiguously specifies the expectations, responsibilities, work products / deliverables and liabilities of both the supplier(s) and the acquirer.

Outcomes:

As a result of successful implementation of the process:

- 1) a contract / agreement will be negotiated, reviewed, approved and awarded to the supplier(s);
- 2) mechanisms for monitoring the capability and performance of the supplier(s) and for mitigation of identified risks will be reviewed and considered for inclusion in the contract conditions;
- 3) proposers / tenderers will be notified of the result of proposal / tender selection.

H.1.12 Supplier Monitoring

Purpose:

The purpose of the *Supplier Monitoring process* is to monitor and facilitate the integration of the supplier's activities in the conduct of the acquisition project in accordance with the relevant requirements and management approaches.

Outcomes:

As a result of successful implementation of the process:

- 1) joint activities will be conducted between the acquirer and the supplier as needed;
- 2) information and data on progress will be exchanged regularly with the supplier;
- 3) performance of the supplier will be monitored against agreed requirements;
- 4) problems will be recorded and tracked to resolution.

H.1.13 Acceptance

Purpose:

The purpose of the *Acceptance process* is to approve and accept the constituted product based against the acceptance criteria. The process will involve a planned and integrated approach that reduces duplication of activities between supplier and acquirer.

Outcomes:

As a result of successful implementation of the process:

- 1) validation and/or verification will be performed against a planned and documented acceptance strategy;
- 2) acceptance will be performed based on the acquisition strategy and conducted according to agreed requirements;
- 3) the delivered product will be evaluated against agreed requirements;
- 4) warranty details, where appropriate, will be confirmed.

NOTE ISO/IEC 14598 may be a suitable basis for product evaluation.

H.1.14 Contract Closure

Purpose:

The purpose of the *Contract Closure process* is to ensure comprehensive information pertaining to the execution and finalisation of the project are collected and co-ordinated across all affected groups.

Outcomes:

As a result of successful implementation of the process:

- 1) finalisation of payments and scheduling of future payments will be agreed;
- 2) securing or return of confidential information provided by the supplier and acquirer will be confirmed;
- 3) exchange of acquisition information results amongst affected group will be effected;
- 4) results of contract, project, technical and financial aspects of the project will be assessed against original requirements and/or objectives;
- 5) the performance of all affected groups will be reviewed;
- 6) relevant project information will be archived in a manner accessible for future acquisitions and improvements.

H.1.15 Supplier Relationships

Purpose:

The purpose of the *Supplier Relationships process* is to improve acquirer-supplier relationships in terms of quality of services and value for money so as to gain a better understanding of the needs of both parties.

Outcomes:

As a result of successful implementation of the process:

- 1) relationships with suppliers which are relevant to current and future needs will be established;
- 2) ownership and co-ordination of relationships will be defined;
- 3) clear understandings of the relationships which are most important in achieving business objectives will be evident;
- 4) potential benefits of improved relationships and reciprocal risks of no change will be identified;
- 5) continuous effectiveness of supplier relationships will be reviewed and monitored.

H.1.16 User relationships

Purpose:

The purpose of the *User Relationships process* is to improve acquirer-user relationships in terms of quality of services and value for money so as to gain a better understanding of the needs of both parties.

Outcomes:

As a result of successful implementation of the process:

- 1) ownership and co-ordination of relationships will be defined;
- 2) clear understandings of the relationships which are most important in achieving business objectives will be evident;
- 3) potential benefits of improved relationships and reciprocal risks of no change will be identified;
- 4) continuous effectiveness of user relationships will be reviewed and monitored.

H.1.17 Financial Management**Purpose:**

The purpose of the *Financial Management process* is to ensure that the costs and budgets for acquisitions are identified and managed in line with agreed plans and objectives.

Outcomes:

As a result of successful implementation of the process:

- 1) financial plans and objectives will be established and maintained;
- 2) budgets will be prepared and approved;
- 3) records will be maintained to satisfy financial audit requirements;
- 4) actual project expenditures will be advised to those responsible for managing projects;
- 5) variances between planned and actual expenditures will be reported and analysed;
- 6) decisions will be taken to ensure financial objectives are met by the responsible personnel.

H.2 Acquisition Process Activities and Tasks**H.2.1 Acquisition Process**

List of activities. The following activities are added to the Acquisition Process:

- 1) Contract closure
- 2) Acquisition policy
- 3) Manage supplier relationships
- 4) Manage user relationships
- 5) Financial management

H.2.1.1 Contract Closure

5.1.6 Contract Closure. This activity consists of the following tasks:

In addition to the normal project management closure activities defined in clause 7.1.5 the acquirer will ensure the following are satisfied :

- a) The finalization of payments is agreed and scheduled;
- b) All confidential information provided to the supplier will be confirmed as secure;
- c) Acquisition information exchange is effected amongst all relevant parties;
- d) The overall results of the contract, project, technical and financial aspects of the acquisition project are assessed against the original requirements and/or objectives.

H.2.1.2 Acquisition Policy

5.1.7 Acquisition Policy. This activity consists of the following tasks :

5.1.7.1 The acquirer will establish the need to deploy a common policy for acquisition across the organization. The acquisition policy should consider the common high level goals and basis for acquisition needs and methods to be deployed on acquisition projects.

5.1.7.2 In defining an effective acquisition policy the following will be considered :

- a) The basis of, or preference for, technology, process, methods, vendors, standards and legally enforceable regulations to optimize acquisitions;
- b) The resources, competencies and skills needed to manage acquisitions including contractual, technical, financial, legal and project management skills;
- c) The standards of quality to be defined;
- d) Relationships towards suppliers, users and other affected parties.

H.2.1.3 Manage Supplier Relationships

5.1.8 Manage Supplier Relationships. This activity consists of the following tasks:

5.1.8.1 The acquisition function within the organization will define a policy with respect to overall relationships with suppliers relevant to its current and future needs. The overall aim to improve acquirer-supplier relationships in terms of service and value for money so as to gain a better understanding of the needs of both parties.

5.1.8.2 It is recognized that in some contracting situations, especially in government or defence sectors, the policy may be a hands off relationship with suppliers, but in most industries there is a move towards strategic relationships with suppliers especially with the advent of electronic procurement.

5.1.8.3 As part of defining a policy the following will be considered:

- a) National or international procurement regulations and/or policies;
- b) Ownership and co-ordination of relationships;
- c) Potential benefits of improved relationships and reciprocal risks of no change;
- d) Review and monitoring of the effectiveness of supplier relationships.

H.2.1.4 Manage User Relationships

5.1.9 Manage User Relationships. This activity consists of the following tasks :

5.1.9.1 The acquisition function within the organization will define a policy with respect to overall relationships with users relevant to its current and future needs. The overall aim to improve acquirer-user relationships in terms of service and value for money so as to gain a better understanding of the needs of both parties.

5.1.9.2 As part of defining a policy the following will be considered:

- a) Ownership and co-ordination of relationships;
- b) Potential benefits of improved relationships and reciprocal risks of no change;
- c) Review and monitoring of the effectiveness of supplier relationships.

H.2.1.5 Financial Management

5.1.10 Financial Management. This activity consists of the following tasks:

5.1.10.1 The organization must ensure sound financial management over acquisition projects. The overall aim is to ensure that the costs and budgets for acquisitions are identified and managed in line with agreed plans and objectives. Financial management often has split responsibilities between different functions within the organization.

5.1.10.2 In order to achieve sound financial management the following will be accomplished:

- a) Financial plans and objectives will be established and maintained;
- b) Budgets will be prepared and approved;
- c) Records will be maintained;
- d) Project expenditures will be advised to those responsible for managing projects;
- e) Variances between planned and actual expenditures will be reported and analyzed;
- f) Decisions will be taken to ensure financial objectives are met.

ICS 35.080

Price based on 53 pages

© ISO/IEC 2002 – All rights reserved

INTERNATIONAL STANDARD

ISO/IEC 12207

First edition
1995-08-01

Information technology — Software life cycle processes

Technologies de l'information — Processus du cycle de vie des logiciels



Reference number
ISO/IEC 12207:1995(E)

ISO/IEC 12207 : 1995 (E)

Contents

Foreword	iii
Introduction	iv
1 Scope	1
2 Normative references	2
3 Definitions	3
4 Application of this International Standard	6
5 Primary life cycle processes	9
5.1 Acquisition process	10
5.2 Supply process	13
5.3 Development process	16
5.4 Operation process	23
5.5 Maintenance process	24
6 Supporting life cycle processes	27
6.1 Documentation process	28
6.2 Configuration management process	29
6.3 Quality assurance process	31
6.4 Verification process	33
6.5 Validation process	36
6.6 Joint review process	38
6.7 Audit process	40
6.8 Problem resolution process	41
7 Organizational life cycle processes	42
7.1 Management process	43
7.2 Infrastructure process	45
7.3 Improvement process	46
7.4 Training process	47
Annexes	
A - Tailoring process	48
B - Guidance on tailoring	49
C - Guidance on processes and organizations	53
D - Bibliography	57

© ISO/IEC 1995

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

ISO/IEC Copyright Office • Case Postale 56 • CH-1211 Genève 20 • Switzerland
Printed in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 12207 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software engineering*.

Annex A forms an integral part of this International Standard. Annexes B and C are for information only.

Introduction

Software is an integral part of information technology and conventional systems, such as transportation, military, medical care, and finance. There is a proliferation of standards, procedures, methods, tools, and environments for developing and managing software. This proliferation has created difficulties in software management and engineering, especially in integrating products and services. The software discipline needs to migrate from this proliferation to a common framework that can be used by software practitioners to "speak the same language" to create and manage software. *This International Standard provides such a common framework.*

The framework covers the life cycle of software from conceptualization of ideas through retirement and consists of processes for acquiring and supplying software products and services. In addition, the framework provides for controlling and improving these processes.

The processes in this International Standard form a comprehensive set. An organization, depending on its purpose, can select an appropriate subset to fulfill that purpose. This International Standard is, therefore, designed to be tailored for an individual organization, project, or application. It is also designed to be used when software is a stand-alone entity, or an embedded or integral part of the total system.

Information technology – Software life cycle processes

1 Scope

1.1 Purpose

This International Standard establishes a common framework for software life cycle processes, with well-defined terminology, that can be referenced by the software industry. It contains processes, activities, and tasks that are to be applied during the acquisition of a system that contains software, a stand-alone software product, and software service and during the supply, development, operation, and maintenance of software products. Software includes the software portion of firmware.

This International Standard also provides a process that can be employed for defining, controlling, and improving software life cycle processes.

1.2 Field of application

This International Standard applies to the acquisition of systems and software products and services, to the supply, development, operation, and maintenance of software products, and to the software portion of firmware, whether performed internally or externally to an organization. Those aspects of system definition needed to provide the context for software products and services are included.

NOTE – The processes used during the software life cycle need to be compatible with the processes used during the system life cycle.

This International Standard is intended for use in a two-party situation and may be equally applied where the two parties are from the same organization. The situation may range from an informal agreement up to a legally binding contract. This International Standard may be used by a single party as self-imposed tasks.

This International Standard is not intended for off-the-shelf software products unless incorporated into a deliverable product.

This International Standard is written for acquirers of systems and software products and services and for suppliers, developers, operators, maintainers, managers, quality assurance managers, and users of software products.

1.3 Tailoring of this International Standard

This International Standard contains a set of processes, activities, and tasks designed to be tailored in respect of software projects. The tailoring process is deletion of non-applicable processes, activities, and tasks.

NOTE – Addition of unique or special processes, activities, and tasks may be provided in the contract.

1.4 Compliance

Compliance with this International Standard is defined as the performance of all the processes, activities, and tasks selected from this International Standard in the Tailoring Process (annex A) for the software project. The performance of a process or an activity is complete when all its required tasks are performed in accordance with the pre-established criteria and the requirements specified in the contract as applicable.

Any organization (for example, national, industrial association, company) imposing this International Standard, as a condition of trade, is responsible for specifying and making public the minimum set of required processes, activities, and tasks, which constitute suppliers' compliance with this International Standard.

1.5 Limitations

This International Standard describes the architecture of the software life cycle processes but does not specify the details of how to implement or perform the activities and tasks included in the processes.

This International Standard is not intended to prescribe the name, format, or explicit content of the documentation to be produced. This International Standard may require development of documents of similar class or type; various plans are an example. This International Standard, however, does not imply that such documents be developed or packaged separately or combined in some fashion. These decisions are left to the user of this International Standard.

This International Standard does not prescribe a specific life cycle model or software development method. The parties of this International Standard are responsible for selecting a life cycle model for the software project and mapping the processes, activities, and tasks in this International Standard onto that model. The parties are also responsible for selecting and applying the software development methods and for performing the activities and tasks suitable for the software project.

This International Standard is not intended to be in conflict with any organization's policies, standards or procedures that are already in place. However, any conflict needs to be resolved and any overriding conditions and situations need to be cited in writing as exceptions to the application of this International Standard.

Throughout this International Standard, "shall" is used to express a provision that is binding between two or more parties, "will" to express a declaration of purpose or intent by one party, "should" to express a recommendation among other possibilities, and "may" to indicate a course of action permissible within the limits of this International Standard.

In this International Standard, there are a number of lists for tasks; none of these is presumed to be exhaustive -- they are intended as examples.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this International Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/AFNOR: 1989, *Dictionary of computer science*.

ISO/IEC 2382-1: 1993, *Information technology – Vocabulary – Part 1: Fundamental terms*.

ISO/IEC 2382-20: 1990, *Information technology – Vocabulary – Part 20: System development*.

ISO 8402: 1994, *Quality management and quality assurance – Vocabulary*.

ISO 9001: 1994, *Quality systems – Model for quality assurance in design, development, production, installation and servicing*.

ISO/IEC 9126: 1991, *Information technology – Software product evaluation – Quality characteristics and guidelines for their use*.

3 Definitions

For the purposes of this International Standard, the definitions given in ISO 8402, ISO/IEC 2382-1 and ISO/IEC 2382-20 apply, together with the following definitions.

NOTE – A product may be interpreted as a part of a system as applicable.

3.1 Acquirer: An organization that acquires or procures a system, software product or software service from a supplier.

NOTE – The acquirer could be one of the following: buyer, customer, owner, user, purchaser.

3.2 Acquisition: The process of obtaining a system, software product or software service.

3.3 Agreement: The definition of terms and conditions under which a working relationship will be conducted.

3.4 Audit: Conducted by an authorized person for the purpose of providing an independent assessment of software products and processes in order to assess compliance with requirements.

3.5 Baseline: A formally approved version of a configuration item, regardless of media, formally designated and fixed at a specific time during the configuration item's life cycle.

3.6 Configuration item: An entity within a configuration that satisfies an end use function and that can be uniquely identified at a given reference point.

3.7 Contract: A binding agreement between two parties, especially enforceable by law, or a similar internal agreement wholly within an organization, for the supply of software service or for the supply, development, production, operation, or maintenance of a software product.

3.8 Developer: An organization that performs development activities (including requirements analysis, design, testing through acceptance) during the software life cycle process.

3.9 Evaluation: A systematic determination of the extent to which an entity meets its specified criteria.

3.10 Firmware: The combination of a hardware device and computer instructions or computer data that reside as read-only software on the hardware device. The software cannot be readily modified under program control.

3.11 Life cycle model: A framework containing the processes, activities, and tasks involved in the development, operation, and maintenance of a software product, spanning the life of the system from the definition of its requirements to the termination of its use.

3.12 Maintainer: An organization that performs maintenance activities.

3.13 Monitoring: An examination of the status of the activities of a supplier and of their results by the acquirer or a third party.

3.14 Non-deliverable item: Hardware or software product that is not required to be delivered under the contract but may be employed in the development of a software product.

3.15 Off-the-shelf product: Product that is already developed and available, usable either "as is" or with modification.

3.16 Operator: An organization that operates the system.

3.17 Process: A set of interrelated activities, which transform inputs into outputs.

NOTE – The term "activities" covers use of resources. [See ISO 8402: 1994, 1.2.]

3.18 Qualification: The process of demonstrating whether an entity is capable of fulfilling specified requirements. [See ISO 8402: 1994, 2.13.]

3.19 Qualification requirement: A set of criteria or conditions that have to be met in order to qualify a software product as complying with its specifications and being ready for use in its target environment.

3.20 Qualification testing: Testing, conducted by the developer and witnessed by the acquirer (as appropriate), to demonstrate that a software product meets its specifications and is ready for use in its target environment.

3.21 Quality assurance: All the planned and systematic activities implemented within the quality system, and demonstrated as needed, to provide adequate confidence that an entity will fulfil requirements for quality.

NOTES

1 There are both internal and external purposes for quality assurance:

- a) Internal quality assurance: within an organization, quality assurance provides confidence to management;
- b) External quality assurance: in contractual situations, quality assurance provides confidence to the customer or others.

2 Some quality control and quality assurance actions are interrelated.

3 Unless requirements for quality fully reflect the needs of the user, quality assurance may not provide adequate confidence.

[ISO 8402: 1994, 3.5]

3.22 Release: A particular version of a configuration item that is made available for a specific purpose (for example, test release).

3.23 Request for proposal [tender]: A document used by the acquirer as the means to announce its intention to potential bidders to acquire a specified system, software product or software service.

3.24 Retirement: Withdrawal of active support by the operation and maintenance organization, partial or total replacement by a new system, or installation of an upgraded system.

3.25 Security: The protection of information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them.

3.26 Software product: The set of computer programs, procedures, and possibly associated documentation and data.

3.27 Software service: Performance of activities, work, or duties connected with a software product, such as its development, maintenance, and operation.

3.28 Software unit: A separately compilable piece of code.

3.29 Statement of work: A document used by the acquirer as the means to describe and specify the tasks to be performed under the contract.

3.30 Supplier: An organization that enters into a contract with the acquirer for the supply of a system, software product or software service under the terms of the contract.

NOTES

1 The term "supplier" is synonymous with contractor, producer, seller, or vendor.

2 The acquirer may designate a part of its organization as supplier.

3.31 System: An integrated composite that consists of one or more of the processes, hardware, software, facilities and people, that provides a capability to satisfy a stated need or objective.

3.32 Test coverage: The extent to which the test cases test the requirements for the system or software product.

3.33 Testability: The extent to which an objective and feasible test can be designed to determine whether a requirement is met.

3.34 User: An individual or organization that uses the operational system to perform a specific function.

NOTE – The user may perform other roles, such as acquirer, developer, or maintainer.

3.35 Validation: Confirmation by examination and provision of objective evidence that the particular requirements for a specific intended use are fulfilled.

NOTES

1 In design and development, validation concerns the process of examining a product to determine conformity with user needs.

2 Validation is normally performed on the final product under defined operating conditions. It may be necessary in earlier stages.

3 "Validated" is used to designate the corresponding status.

4 Multiple validations may be carried out if there are different intended uses.

[ISO 8402: 1994, 2.18]

3.36 Verification: Confirmation by examination and provision of objective evidence that specified requirements have been fulfilled.

NOTES

1 In design and development, verification concerns the process of examining the result of a given activity to determine conformity with the stated requirement for that activity.

2 "Verified" is used to designate the corresponding status.

[ISO 8402: 1994, 2.17]

3.37 Version: An identified instance of an item.

NOTE – Modification to a version of a software product, resulting in a new version, requires configuration management action.

4 Application of this International Standard

This clause presents the software life cycle processes that can be employed to acquire, supply, develop, operate, and maintain software products. The objective is to provide a road map for the users of this International Standard so that they can orient themselves in it and apply it judiciously.

4.1 Organization of this International Standard

4.1.1 Life cycle processes

This International Standard groups the activities that may be performed during the life cycle of software into five primary processes, eight supporting processes, and four organizational processes. Each life cycle process is divided into a set of activities; each activity is further divided into a set of tasks. Subclause numbering a.b denotes a process, a.b.c an activity, and a.b.c.d a task. These life cycle processes are introduced below and depicted in figure 1.

4.1.1.1 Primary life cycle processes

The primary life cycle processes (clause 5) consist of five processes that serve primary parties during the life cycle of software. A primary party is one that initiates or performs the development, operation, or maintenance of software products. These primary parties are the acquirer, the supplier, the developer, the operator, and the maintainer of software products. The primary processes are:

- 1) *Acquisition process* (subclause 5.1). Defines the activities of the acquirer, the organization that acquires a system, software product or software service.
- 2) *Supply process* (subclause 5.2). Defines the activities of the supplier, the organization that provides the system, software product or software service to the acquirer.
- 3) *Development process* (subclause 5.3). Defines the activities of the developer, the organization that defines and develops the software product.
- 4) *Operation process* (subclause 5.4). Defines the activities of the operator, the organization that provides the service of operating a computer system in its live environment for its users.
- 5) *Maintenance process* (subclause 5.5). Defines the activities of the maintainer, the organization that provides the service of maintaining the software product; that is, managing modifications to the software product to keep it current and in operational fitness. This process includes the migration and retirement of the software product.

4.1.1.2 Supporting life cycle processes

The supporting life cycle processes (clause 6) consist of eight processes. A supporting process supports another process as an integral part with a distinct purpose and contributes to the success and quality of the software project. A supporting process is employed and executed, as needed, by another process. The supporting processes are:

- 1) *Documentation process* (subclause 6.1). Defines the activities for recording the information produced by a life cycle process.
- 2) *Configuration management process* (subclause 6.2). Defines the configuration management activities.
- 3) *Quality assurance process* (subclause 6.3). Defines the activities for objectively assuring that the software products and processes are in conformance with their specified requirements and adhere to their established plans. Joint Reviews, Audits, Verification, and Validation may be used as techniques of Quality Assurance.

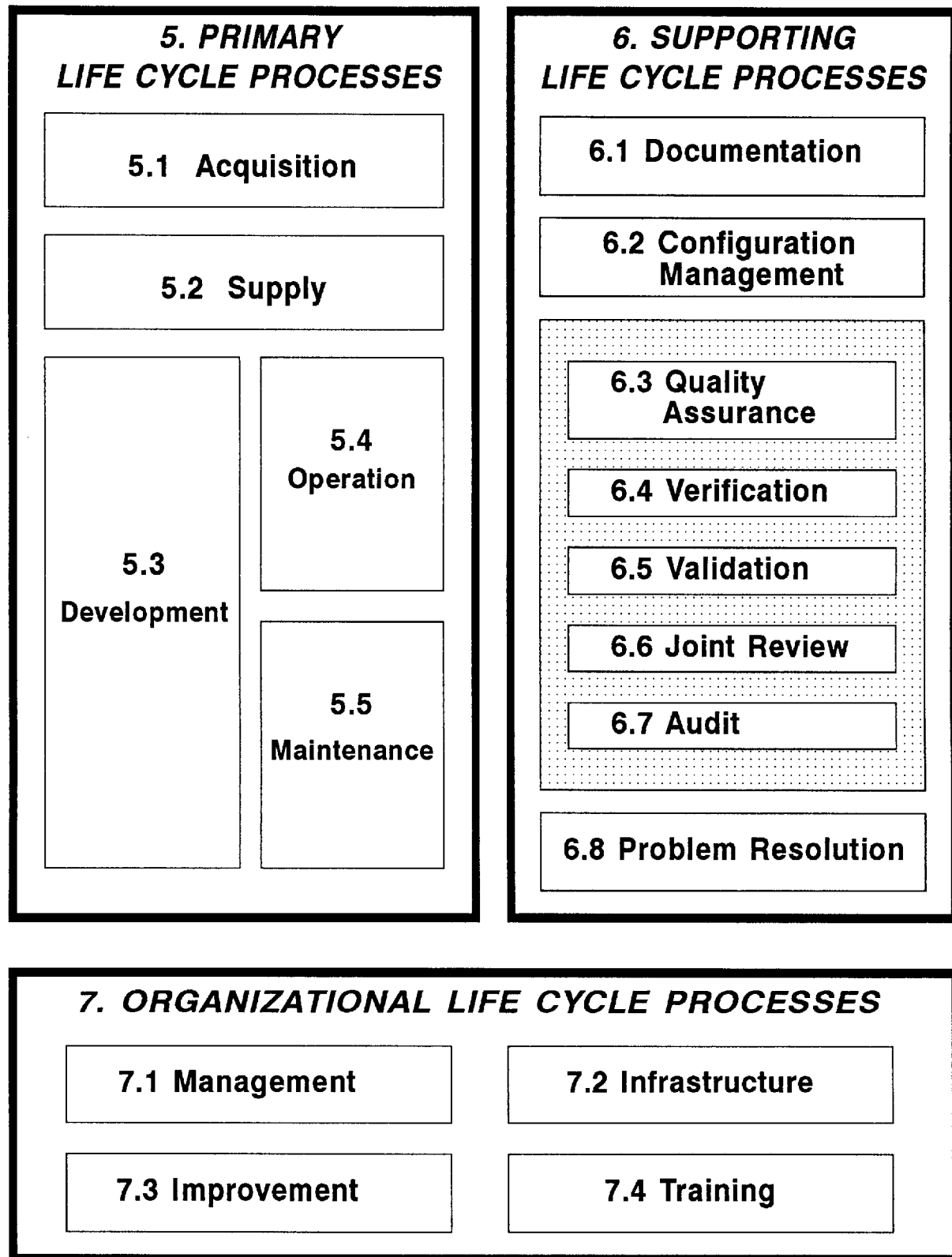


Figure 1. Structure of the International Standard

- 4) *Verification process* (subclause 6.4). Defines the activities (for the acquirer, the supplier, or an independent party) for verifying the software products in varying depth depending on the software project.
- 5) *Validation process* (subclause 6.5). Defines the activities (for the acquirer, the supplier, or an independent party) for validating the software products of the software project.
- 6) *Joint review process* (subclause 6.6). Defines the activities for evaluating the status and products of an activity. This process may be employed by any two parties, where one party (reviewing party) reviews another party (reviewed party) in a joint forum.
- 7) *Audit process* (subclause 6.7). Defines the activities for determining compliance with the requirements, plans and contract. This process may be employed by any two parties, where one party (auditing party) audits the software products or activities of another party (audited party).
- 8) *Problem resolution process* (subclause 6.8). Defines a process for analyzing and removing the problems (including non-conformances), whatever their nature or source, that are discovered during the execution of development, operation, maintenance, or other processes.

4.1.1.3 Organizational life cycle processes

The organizational life cycle processes (clause 7) consist of four processes. They are employed by an organization to establish and implement an underlying structure made up of associated life cycle processes and personnel and continuously improve the structure and processes. They are typically employed outside the realm of specific projects and contracts; however, lessons from such projects and contracts contribute to the improvement of the organization. The organizational processes are:

- 1) *Management process* (subclause 7.1). Defines the basic activities of the management, including project management, during a life cycle process.
- 2) *Infrastructure process* (subclause 7.2). Defines the basic activities for establishing the underlying structure of a life cycle process.
- 3) *Improvement process* (subclause 7.3). Defines the basic activities that an organization (that is, acquirer, supplier, developer, operator, maintainer, or the manager of another process) performs for establishing, measuring, controlling, and improving its life cycle process.
- 4) *Training process* (subclause 7.4). Defines the activities for providing adequately trained personnel.

4.1.2 Tailoring process. Annex A, which is normative, defines the basic activities needed to perform tailoring of this International Standard. Annex B contains a brief guidance on tailoring the requirements of this International Standard; it lists the key factors upon which tailoring decisions may be made.

4.1.3 Relationship between the processes and organizations

This International Standard contains various processes that are applied throughout the life cycle of software by various organizations depending on their needs and goals. For understandability, annex C presents the relationships between the life cycle processes and related parties.

5 Primary life cycle processes

This clause defines the following primary life cycle processes:

- 1) Acquisition process;
- 2) Supply process;
- 3) Development process;
- 4) Operation process;
- 5) Maintenance process.

The activities and tasks in a primary process are the responsibility of the organization initiating and performing that process. This organization ensures that the process is in existence and functional.

5.1 Acquisition process

The Acquisition Process contains the activities and tasks of the acquirer. The process begins with the definition of the need to acquire a system, software product or software service. The process continues with the preparation and issue of a request for proposal, selection of a supplier, and management of the acquisition process through to the acceptance of the system, software product or software service.

The individual organization having the need may be called the owner. The owner may contract any or all of the acquisition activities to an agent who will in turn conduct these activities according to the Acquisition Process. The acquirer in this subclause may be the owner or the agent.

The acquirer manages the Acquisition Process at the project level following the Management Process (7.1), which is instantiated in this process; establishes an infrastructure under the process following the Infrastructure Process (7.2); tailors the process for the project following the Tailoring Process (annex A); and manages the process at the organizational level following the Improvement Process (7.3) and the Training Process (7.4).

List of activities: This process consists of the following activities:

- 1) Initiation;
- 2) Request-for-Proposal [-tender] preparation;
- 3) Contract preparation and update;
- 4) Supplier monitoring;
- 5) Acceptance and completion.

5.1.1 Initiation. This activity consists of the following tasks:

5.1.1.1 The acquirer begins the acquisition process by describing a concept or a need to acquire, develop, or enhance a system, software product or software service.

5.1.1.2 The acquirer will define and analyze the system requirements. The system requirements should include business, organizational and user as well as safety, security, and other criticality requirements along with related design, testing, and compliance standards and procedures.

5.1.1.3 If the acquirer retains a supplier to perform system requirements analysis, the acquirer will approve the analyzed requirements.

5.1.1.4 The acquirer may perform the definition and analysis of software requirements by itself or may retain a supplier to perform this task.

5.1.1.5 The Development Process (5.3) should be used to perform the tasks in 5.1.1.2 and 5.1.1.4.

5.1.1.6 The acquirer will consider options for acquisition against analysis of appropriate criteria to include risk, cost and benefits for each option. Options include:

- a) Purchase an off-the-shelf software product that satisfies the requirements.
- b) Develop the software product or obtain the software service internally.
- c) Develop the software product or obtain the software service through contract.
- d) A combination of a, b, and c above.
- e) Enhance an existing software product or service.

5.1.1.7 When an off-the-shelf software product is to be acquired, the acquirer will ensure the following conditions are satisfied:

- a) The requirements for the software product are satisfied.
- b) The documentation is available.
- c) Proprietary, usage, ownership, warranty and licensing rights are satisfied.
- d) Future support for the software product is planned.

5.1.1.8 The acquirer should prepare, document and execute an acquisition plan. The plan should contain the following:

- a) Requirements for the system;
- b) Planned employment of the system;
- c) Type of contract to be employed;
- d) Responsibilities of the organizations involved;
- e) Support concept to be used;
- f) Risks considered as well as methods to manage the risks.

5.1.1.9 The acquirer should define and document the acceptance strategy and conditions (criteria).

5.1.2 Request-for-proposal [-tender] preparation. This activity consists of the following tasks:

5.1.2.1 The acquirer should document the acquisition requirements (e.g., request for proposal), the content of which depends upon the acquisition option selected in 5.1.1.6. The acquisition documentation should include, as appropriate:

- a) System requirements;
- b) Scope statement;
- c) Instructions for bidders;
- d) List of software products;
- e) Terms and conditions;
- f) Control of subcontracts;
- g) Technical constraints (e.g., target environment).

5.1.2.2 The acquirer should determine which processes, activities, and tasks of this International Standard are appropriate for the project and should tailor them accordingly. Especially, the acquirer should specify the applicable supporting processes (clause 6) and their performing organizations, including responsibilities (if other than supplier), so that the suppliers may, in their proposals, define the approach to each of the specified supporting processes. The acquirer will define the scope of those tasks that reference the contract.

5.1.2.3 The acquisition documentation will also define the contract milestones at which the supplier's progress will be reviewed and audited as part of monitoring the acquisition (see 6.6 and 6.7).

5.1.2.4 The acquisition requirements should be given to the organization selected for performing the acquisition activities.

5.1.3 Contract preparation and update. This activity consists of the following tasks:

5.1.3.1 The acquirer should establish a procedure for supplier selection including proposal evaluation criteria and requirements compliance weighting.

5.1.3.2 The acquirer should select a supplier based upon the evaluation of the suppliers' proposals, capabilities, and other factors that need to be considered.

5.1.3.3 The acquirer may involve other parties, including potential suppliers, before contract award, in tailoring this International Standard for the project. However, the acquirer will make the final decision on the tailoring. The acquirer will include or reference the tailored International Standard in the contract.

5.1.3.4 The acquirer will then prepare and negotiate a contract with the supplier, that addresses the acquisition requirements, including the cost and schedule, of the software product or service to be delivered. The contract will address proprietary, usage, ownership, warranty and licensing rights associated with the reusable off-the-shelf software products.

5.1.3.5 Once the contract is underway, the acquirer will control changes to the contract through negotiation with the supplier as part of a change control mechanism. Changes to the contract shall be investigated for impact on project plans, costs, benefits, quality, and schedule.

NOTE – The acquirer determines whether the term "contract" or "agreement" is to be used in the application of this International Standard.

5.1.4 Supplier monitoring. This activity consists of the following tasks:

5.1.4.1 The acquirer will monitor the supplier's activities in accordance with the Joint Review Process (6.6) and the Audit Process (6.7). The acquirer should supplement the monitoring with the Verification Process (6.4) and the Validation Process (6.5) as needed.

5.1.4.2 The acquirer will cooperate with the supplier to provide all necessary information in a timely manner and resolve all pending items.

5.1.5 Acceptance and completion. This activity consists of the following tasks:

5.1.5.1 The acquirer should prepare for acceptance based on the defined acceptance strategy and criteria. The preparation of test cases, test data, test procedures, and test environment should be included. The extent of supplier involvement should be defined.

5.1.5.2 The acquirer will conduct acceptance review and acceptance testing of the deliverable software product or service and will accept it from the supplier when all acceptance conditions are satisfied. The acceptance procedure should comply with the provisions of 5.1.1.9.

5.1.5.3 After acceptance, the acquirer should take the responsibility for the configuration management of the delivered software product (see 6.2).

NOTE – The acquirer may install the software product or perform the software service in accordance with instructions defined by the supplier.

5.2 Supply process

The Supply Process contains the activities and tasks of the supplier. The process may be initiated either by a decision to prepare a proposal to answer an acquirer's request for proposal or by signing and entering into a contract with the acquirer to provide the system, software product or software service. The process continues with the determination of procedures and resources needed to manage and assure the project, including development of project plans and execution of the plans through delivery of the system, software product or software service to the acquirer.

The supplier manages the Supply Process at the project level following the Management Process (7.1), which is instantiated in this process; establishes an infrastructure under the process following the Infrastructure Process (7.2); tailors the process for the project following the Tailoring Process (annex A); and manages the process at the organizational level following the Improvement Process (7.3) and the Training Process (7.4).

List of activities: This process consists of the following activities:

- 1) Initiation;
- 2) Preparation of response;
- 3) Contract;
- 4) Planning;
- 5) Execution and control;
- 6) Review and evaluation;
- 7) Delivery and completion.

5.2.1 Initiation. This activity consists of the following tasks:

5.2.1.1 The supplier conducts a review of requirements in the request for proposal taking into account organizational policies and other regulations.

5.2.1.2 The supplier should make a decision to bid or accept the contract.

5.2.2 Preparation of response. This activity consists of the following task:

5.2.2.1 The supplier should define and prepare a proposal in response to the request for proposal, including its recommended tailoring of this International Standard.

5.2.3 Contract. This activity consists of the following tasks:

5.2.3.1 The supplier shall negotiate and enter into a contract with the acquirer organization to provide the software product or service.

5.2.3.2 The supplier may request modification to the contract as part of the change control mechanism.

5.2.4 Planning. This activity consists of the following tasks:

5.2.4.1 The supplier shall conduct a review of the acquisition requirements to define the framework for managing and assuring the project and for assuring the quality of the deliverable software product or service.

5.2.4.2 If not stipulated in the contract, the supplier shall define or select a software life cycle model appropriate to the scope, magnitude, and complexity of the project. The processes, activities, and tasks of this International Standard shall be selected and mapped onto the life cycle model.

5.2.4.3 The supplier shall establish requirements for the plans for managing and assuring the project and for assuring the quality of the deliverable software product or service. Requirements for the plans should include resource needs and acquirer involvement.

5.2.4.4 Once the planning requirements are established, the supplier shall consider the options for developing the software product or providing the software service, against an analysis of risks associated with each option. Options include:

- a) Develop the software product or provide the software service using internal resources.
- b) Develop the software product or provide the software service by subcontracting.
- c) Obtain off-the-shelf software products from internal or external sources.
- d) A combination of a, b, and c above.

5.2.4.5 The supplier shall develop and document project management plan(s) based upon the planning requirements and options selected in 5.2.4.4. Items to be considered in the plan include but are not limited to the following:

- a) Project organizational structure and authority and responsibility of each organizational unit, including external organizations;
- b) Engineering environment (for development, operation, or maintenance, as applicable), including test environment, library, equipment, facilities, standards, procedures, and tools;
- c) Work breakdown structure of the life cycle processes and activities, including the software products, software services and non-deliverable items, to be performed together with budgets, staffing, physical resources, software size, and schedules associated with the tasks;
- d) Management of the quality characteristics of the software products or services. Separate plans for quality may be developed.
- e) Management of the safety, security, and other critical requirements of the software products or services. Separate plans for safety and security may be developed.
- f) Subcontractor management, including subcontractor selection and involvement between the subcontractor and the acquirer, if any;
- g) Quality assurance (see 6.3);
- h) Verification (see 6.4) and validation (see 6.5); including the approach for interfacing with the verification and validation agent, if specified;
- i) Acquirer involvement; that is, by such means as joint reviews (see 6.6), audits (see 6.7), informal meetings, reporting, modification and change; implementation, approval, acceptance, and access to facilities;
- j) User involvement; by such means as requirements setting exercises, prototype demonstrations and evaluations;
- k) Risk management; that is management of the areas of the project that involve potential technical, cost, and schedule risks;
- l) Security policy; that is, the rules for need-to-know and access-to-information at each project organization level;
- m) Approval required by such means as regulations, required certifications, proprietary, usage, ownership, warranty and licensing rights;
- n) Means for scheduling, tracking, and reporting;
- o) Training of personnel (see 7.4).

5.2.5 Execution and control. This activity consists of the following tasks:

5.2.5.1 The supplier shall implement and execute the project management plan(s) developed in 5.2.4.

5.2.5.2 The supplier shall:

- a) Develop the software product in accordance with Development Process (5.3).
- b) Operate the software product in accordance with Operation Process (5.4).
- c) Maintain the software product in accordance with Maintenance Process (5.5).

5.2.5.3 The supplier shall monitor and control the progress and the quality of the software products or services of the project throughout the contracted life cycle. This shall be an ongoing, iterative task, which shall provide for:

- a) Monitoring progress of technical performance, costs, and schedules and reporting of project status;
- b) Problem identification, recording, analysis, and resolution.

5.2.5.4 The supplier shall manage and control the subcontractors in accordance with the Acquisition Process (5.1). The supplier shall pass down all contractual requirements necessary to ensure that the software product or service delivered to the acquirer is developed or performed in accordance with the prime-contract requirements.

5.2.5.5 The supplier shall interface with the independent verification, validation, or test agent as specified in the contract and project plans.

5.2.5.6 The supplier shall interface with other parties as specified in the contract and project plans.

5.2.6 Review and evaluation. This activity consists of the following tasks:

5.2.6.1 The supplier should coordinate contract review activities, interfaces, and communication with the acquirer's organization.

5.2.6.2 The supplier shall conduct or support the informal meetings, acceptance review, acceptance testing, joint reviews, and audits with the acquirer as specified in the contract and project plans. The joint reviews shall be conducted in accordance with 6.6, audits in accordance with 6.7.

5.2.6.3 The supplier shall perform verification and validation in accordance with 6.4 and 6.5 respectively to demonstrate that the software products or services and processes fully satisfy their respective requirements.

5.2.6.4 The supplier shall make available to the acquirer the reports of evaluation, reviews, audits, testing, and problem resolutions as specified in the contract.

5.2.6.5 The supplier shall provide the acquirer access to the supplier's and subcontractors' facilities for review of software products or services as specified in the contract and project plans.

5.2.6.6 The supplier shall perform quality assurance activities in accordance with 6.3.

5.2.7 Delivery and completion. This activity consists of the following tasks:

5.2.7.1 The supplier shall deliver the software product or service as specified in the contract.

5.2.7.2 The supplier shall provide assistance to the acquirer in support of the delivered software product or service as specified in the contract.

5.3 Development process

The Development Process contains the activities and tasks of the developer. The process contains the activities for requirements analysis, design, coding, integration, testing, and installation and acceptance related to software products. It may contain system related activities if stipulated in the contract. The developer performs or supports the activities in this process in accordance with the contract.

The developer manages the Development Process at the project level following the Management Process (7.1), which is instantiated in this process; establishes an infrastructure under the process following the Infrastructure Process (7.2); tailors the process for the project following the Tailoring Process (annex A); and manages the process at the organizational level following the Improvement Process (7.3) and the Training Process (7.4). When the developer is the supplier of the developed software product, the developer performs the Supply Process (5.2).

List of activities: This process consists of the following activities:

- 1) Process implementation;
- 2) System requirements analysis;
- 3) System architectural design;
- 4) Software requirements analysis;
- 5) Software architectural design;
- 6) Software detailed design;
- 7) Software coding and testing;
- 8) Software integration;
- 9) Software qualification testing;
- 10) System integration;
- 11) System qualification testing;
- 12) Software installation;
- 13) Software acceptance support.

5.3.1 Process implementation. This activity consists of the following tasks:

5.3.1.1 If not stipulated in the contract, the developer shall define or select a software life cycle model appropriate to the scope, magnitude, and complexity of the project. The activities and tasks of the Development Process shall be selected and mapped onto the life cycle model.

NOTE – These activities and tasks may overlap or interact and may be performed iteratively or recursively.

5.3.1.2 The developer shall:

- a) Document the outputs in accordance with the Documentation Process (6.1).
- b) Place the outputs under the Configuration Management Process (6.2) and perform change control in accordance with it.
- c) Document and resolve problems and non-conformances found in the software products and tasks in accordance with the Problem Resolution Process (6.8).
- d) Perform the supporting processes (clause 6) as specified in the contract.

5.3.1.3 The developer shall select, tailor, and use those standards, methods, tools, and computer programming languages (if not stipulated in the contract) that are documented, appropriate, and established by the organization for performing the activities of the Development Process and supporting processes (clause 6).

5.3.1.4 The developer shall develop plans for conducting the activities of the development process. The plans should include specific standards, methods, tools, actions, and responsibility associated with the development and qualification of all requirements including safety and security. If necessary, separate plans may be developed. These plans shall be documented and executed.

5.3.1.5 Non-deliverable items may be employed in the development or maintenance of the software product. However, it shall be ensured that the operation and maintenance of the deliverable software product after its delivery to the acquirer are independent of such items, otherwise those items should be considered as deliverable.

5.3.2 System requirements analysis. This activity consists of the following tasks, which the developer shall perform or support as required by the contract:

5.3.2.1 The specific intended use of the system to be developed shall be analyzed to specify system requirements. The system requirements specification shall describe: functions and capabilities of the system; business, organizational and user requirements; safety, security, human-factors engineering (ergonomics), interface, operations, and maintenance requirements; design constraints and qualification requirements. The system requirements specification shall be documented.

5.3.2.2 The system requirements shall be evaluated considering the criteria listed below. The results of evaluations shall be documented.

- a) Traceability to acquisition needs;
- b) Consistency with acquisition needs;
- c) Testability;
- d) Feasibility of system architectural design;
- e) Feasibility of operation and maintenance.

5.3.3 System architectural design. This activity consists of the following tasks, which the developer shall perform or support as required by the contract:

5.3.3.1 A top-level architecture of the system shall be established. The architecture shall identify items of hardware, software, and manual-operations. It shall be ensured that all the system requirements are allocated among the items. Hardware configuration items, software configuration items, and manual operations shall be subsequently identified from these items. The system architecture and the system requirements allocated to the items shall be documented.

5.3.3.2 The system architecture and the requirements for the items shall be evaluated considering the criteria listed below. The results of the evaluations shall be documented.

- a) Traceability to the system requirements;
- b) Consistency with the system requirements;
- c) Appropriateness of design standards and methods used;
- d) Feasibility of the software items fulfilling their allocated requirements;
- e) Feasibility of operation and maintenance.

5.3.4 Software requirements analysis. For each software item (or software configuration item, if identified), this activity consists of the following tasks:

5.3.4.1 The developer shall establish and document software requirements, including the quality characteristics specifications, described below. Guidance for specifying quality characteristics may be found in ISO/IEC 9126.

- a) Functional and capability specifications, including performance, physical characteristics, and environmental conditions under which the software item is to perform;
- b) Interfaces external to the software item;
- c) Qualification requirements;
- d) Safety specifications, including those related to methods of operation and maintenance, environmental influences, and personnel injury;
- e) Security specifications, including those related to compromise of sensitive information;

- f) Human-factors engineering (ergonomics) specifications, including those related to manual operations, human-equipment interactions, constraints on personnel, and areas needing concentrated human attention, that are sensitive to human errors and training;
- g) Data definition and database requirements;
- h) Installation and acceptance requirements of the delivered software product at the operation and maintenance site(s);
- i) User documentation;
- j) User operation and execution requirements;
- k) User maintenance requirements.

5.3.4.2 The developer shall evaluate the software requirements considering the criteria listed below. The results of the evaluations shall be documented.

- a) Traceability to system requirements and system design;
- b) External consistency with system requirements;
- c) Internal consistency;
- d) Testability;
- e) Feasibility of software design;
- f) Feasibility of operation and maintenance.

5.3.4.3 The developer shall conduct joint review(s) in accordance with 6.6. Upon successful completion of the review(s), a baseline for the requirements of the software item shall be established.

5.3.5 Software architectural design. For each software item (or software configuration item, if identified), this activity consists of the following tasks:

5.3.5.1 The developer shall transform the requirements for the software item into an architecture that describes its top-level structure and identifies the software components. It shall be ensured that all the requirements for the software item are allocated to its software components and further refined to facilitate detailed design. The architecture of the software item shall be documented.

5.3.5.2 The developer shall develop and document a top-level design for the interfaces external to the software item and between the software components of the software item.

5.3.5.3 The developer shall develop and document a top-level design for the database.

5.3.5.4 The developer should develop and document preliminary versions of user documentation.

5.3.5.5 The developer shall define and document preliminary test requirements and the schedule for Software Integration.

5.3.5.6 The developer shall evaluate the architecture of the software item and the interface and database designs considering the criteria listed below. The results of the evaluations shall be documented.

- a) Traceability to the requirements of the software item;
- b) External consistency with the requirements of the software item;
- c) Internal consistency between the software components;
- d) Appropriateness of design methods and standards used;
- e) Feasibility of detailed design;
- f) Feasibility of operation and maintenance.

5.3.5.7 The developer shall conduct joint review(s) in accordance with 6.6.

5.3.6 Software detailed design. For each software item (or software configuration item, if identified), this activity consists of the following tasks:

5.3.6.1 The developer shall develop a detailed design for each software component of the software item. The software components shall be refined into lower levels containing software units that can be coded, compiled, and tested. It shall be ensured that all the software requirements are allocated from the software components to software units. The detailed design shall be documented.

5.3.6.2 The developer shall develop and document a detailed design for the interfaces external to the software item, between the software components, and between the software units. The detailed design of the interfaces shall permit coding without the need for further information.

5.3.6.3 The developer shall develop and document a detailed design for the database.

5.3.6.4 The developer shall update user documentation as necessary.

5.3.6.5 The developer shall define and document test requirements and schedule for testing software units. The test requirements should include stressing the software unit at the limits of its requirements.

5.3.6.6 The developer shall update the test requirements and the schedule for Software Integration.

5.3.6.7 The developer shall evaluate the software detailed design and test requirements considering the criteria listed below. The results of the evaluations shall be documented.

- a) Traceability to the requirements of the software item;
- b) External consistency with architectural design;
- c) Internal consistency between software components and software units;
- d) Appropriateness of design methods and standards used;
- e) Feasibility of testing;
- f) Feasibility of operation and maintenance.

5.3.6.8 The developer shall conduct joint review(s) in accordance with 6.6.

5.3.7 Software coding and testing. For each software item (or software configuration item, if identified), this activity consists of the following tasks:

5.3.7.1 The developer shall develop and document the following:

- a) Each software unit and database;
- b) Test procedures and data for testing each software unit and database.

5.3.7.2 The developer shall test each software unit and database ensuring that it satisfies its requirements. The test results shall be documented.

5.3.7.3 The developer shall update the user documentation as necessary.

5.3.7.4 The developer shall update the test requirements and the schedule for Software Integration.

5.3.7.5 The developer shall evaluate software code and test results considering the criteria listed below. The results of the evaluations shall be documented.

- a) Traceability to the requirements and design of the software item;
- b) External consistency with the requirements and design of the software item;
- c) Internal consistency between unit requirements;
- d) Test coverage of units;
- e) Appropriateness of coding methods and standards used;
- f) Feasibility of software integration and testing;
- g) Feasibility of operation and maintenance.

5.3.8 Software integration. For each software item (or software configuration item, if identified), this activity consists of the following tasks:

5.3.8.1 The developer shall develop an integration plan to integrate the software units and software components into the software item. The plan shall include test requirements, procedures, data, responsibilities, and schedule. The plan shall be documented.

5.3.8.2 The developer shall integrate the software units and software components and test as the aggregates are developed in accordance with the integration plan. It shall be ensured that each aggregate satisfies the requirements of the software item and that the software item is integrated at the conclusion of the integration activity. The integration and test results shall be documented.

5.3.8.3 The developer shall update the user documentation as necessary.

5.3.8.4 The developer shall develop and document, for each qualification requirement of the software item, a set of tests, test cases (inputs, outputs, test criteria), and test procedures for conducting Software Qualification Testing. The developer shall ensure that the integrated software item is ready for Software Qualification Testing.

5.3.8.5 The developer shall evaluate the integration plan, design, code, tests, test results, and user documentation considering the criteria listed below. The results of the evaluations shall be documented.

- a) Traceability to the system requirements;
- b) External consistency with the system requirements;
- c) Internal consistency;
- d) Test coverage of the requirements of the software item;
- e) Appropriateness of test standards and methods used;
- f) Conformance to expected results;
- g) Feasibility of software qualification testing;
- h) Feasibility of operation and maintenance.

5.3.8.6 The developer shall conduct joint review(s) in accordance with 6.6.

5.3.9 Software qualification testing. For each software item (or software configuration item, if identified), this activity consists of the following tasks:

5.3.9.1 The developer shall conduct qualification testing in accordance with the qualification requirements for the software item. It shall be ensured that the implementation of each software requirement is tested for compliance. The qualification testing results shall be documented.

5.3.9.2 The developer shall update the user documentation as necessary.

5.3.9.3 The developer shall evaluate the design, code, tests, test results, and user documentation considering the criteria listed below. The results of the evaluations shall be documented.

- a) Test coverage of the requirements of the software item;
- b) Conformance to expected results;
- c) Feasibility of system integration and testing, if conducted ;
- d) Feasibility of operation and maintenance.

5.3.9.4 The developer shall support audit(s) in accordance with 6.7. The results of the audits shall be documented. If both hardware and software are under development or integration, the audits may be postponed until the System Qualification Testing.

5.3.9.5 Upon successful completion of the audits, if conducted, the developer shall:

- a) Update and prepare the deliverable software product for System Integration, System Qualification Testing, Software Installation, or Software Acceptance Support as applicable.

- b) Establish a baseline for the design and code of the software item.

NOTE – The Software Qualification Testing may be used in the Verification Process (6.4) or the Validation Process (6.5).

5.3.10 System integration. This activity consists of the following tasks, which the developer shall perform or support as required by the contract.

5.3.10.1 The software configuration items shall be integrated, with hardware configuration items, manual operations, and other systems as necessary, into the system. The aggregates shall be tested, as they are developed, against their requirements. The integration and the test results shall be documented.

5.3.10.2 For each qualification requirement of the system, a set of tests, test cases (inputs, outputs, test criteria), and test procedures for conducting System Qualification Testing shall be developed and documented. The developer shall ensure that the integrated system is ready for System Qualification Testing.

5.3.10.3 The integrated system shall be evaluated considering the criteria listed below. The results of the evaluations shall be documented.

- a) Test coverage of system requirements;
- b) Appropriateness of test methods and standards used;
- c) Conformance to expected results;
- d) Feasibility of system qualification testing;
- e) Feasibility of operation and maintenance.

5.3.11 System qualification testing. This activity consists of the following tasks, which the developer shall perform or support as required by the contract.

5.3.11.1 System qualification testing shall be conducted in accordance with the qualification requirements specified for the system. It shall be ensured that the implementation of each system requirement is tested for compliance and that the system is ready for delivery. The qualification testing results shall be documented.

5.3.11.2 The system shall be evaluated considering the criteria listed below. The results of the evaluations shall be documented.

- a) Test coverage of system requirements;
- b) Conformance to expected results;
- c) Feasibility of operation and maintenance.

5.3.11.3 The developer shall support audit(s) in accordance with 6.7. The results of the audit(s) shall be documented.

NOTE – This subclause is not applicable to those software configuration items for which audits were conducted previously.

5.3.11.4 Upon successful completion of the audit(s), if conducted, the developer shall:

- a) Update and prepare the deliverable software product for Software Installation and Software Acceptance Support.
- b) Establish a baseline for the design and code of each software configuration item.

NOTE – The System Qualification Testing may be used in the Verification Process (6.4) or the Validation Process (6.5).

5.3.12 Software installation. This activity consists of the following tasks:

5.3.12.1 The developer shall develop a plan to install the software product in the target environment as designated in the contract. The resources and information necessary to install the software product shall be determined and be available. As specified in the contract, the developer shall assist the acquirer with the set-up activities. Where the installed software product is replacing an existing system, the developer shall support any parallel running activities that are required by contract. The installation plan shall be documented.

5.3.12.2 The developer shall install the software product in accordance with the installation plan. It shall be ensured that the software code and databases initialize, execute, and terminate as specified in the contract. The installation events and results shall be documented.

5.3.13 Software acceptance support. This activity consists of the following tasks:

5.3.13.1 The developer shall support the acquirer's acceptance review and testing of the software product. Acceptance review and testing shall consider the results of the Joint Reviews (6.6), Audits (6.7), Software Qualification Testing, and System Qualification Testing (if performed). The results of the acceptance review and testing shall be documented.

5.3.13.2 The developer shall complete and deliver the software product as specified in the contract.

5.3.13.3 The developer shall provide initial and continuing training and support to the acquirer as specified in the contract.

5.4 Operation process

The Operation Process contains the activities and tasks of the operator. The process covers the operation of the software product and operational support to users. Because operation of software product is integrated into the operation of the system, the activities and tasks of this process refer to the system.

The operator manages the Operation Process at the project level following the Management Process (7.1), which is instantiated in this process; establishes an infrastructure under the process following the Infrastructure Process (7.2); tailors the process for the project following the Tailoring Process (annex A); and manages the process at the organizational level following the Improvement Process (7.3) and the Training Process (7.4). When the operator is the supplier of the operation service, the operator performs the Supply Process (5.2).

List of activities. This process consists of the following activities:

- 1) Process implementation;
- 2) Operational testing;
- 3) System operation;
- 4) User support.

5.4.1 Process implementation. This activity consists of the following tasks:

5.4.1.1 The operator shall develop a plan and set operational standards for performing the activities and tasks of this process. The plan shall be documented and executed.

5.4.1.2 The operator shall establish procedures for receiving, recording, resolving, tracking problems, and providing feedback. Whenever problems are encountered, they shall be recorded and entered into the Problem Resolution Process (6.8).

5.4.1.3 The operator shall establish procedures for testing the software product in its operation environment, for entering problem reports and modification requests to the Maintenance Process (5.5), and for releasing the software product for operational use.

5.4.2 Operational testing. This activity consists of the following tasks:

5.4.2.1 For each release of the software product, the operator shall perform operational testing, and, on satisfying the specified criteria, release the software product for operational use.

5.4.2.2 The operator shall ensure that the software code and databases initialize, execute, and terminate as described in the plan.

5.4.3 System operation. This activity consists of the following task:

5.4.3.1 The system shall be operated in its intended environment according to the user documentation.

5.4.4 User support. This activity consists of the following tasks:

5.4.4.1 The operator shall provide assistance and consultation to the users as requested. These requests and subsequent actions shall be recorded and monitored.

5.4.4.2 The operator shall forward user requests, as necessary, to the Maintenance Process (clause 5.5) for resolution. These requests shall be addressed and the actions that are planned and taken shall be reported to the originators of the requests. All resolutions shall be monitored to conclusion.

5.4.4.3 If a reported problem has a temporary work-around before a permanent solution can be released, the originator of the problem report shall be given the option to use it. Permanent corrections, releases that include previously omitted functions or features, and system improvements shall be applied to the operational software product using the Maintenance Process (5.5).

5.5 Maintenance process

The Maintenance Process contains the activities and tasks of the maintainer. This process is activated when the software product undergoes modifications to code and associated documentation due to a problem or the need for improvement or adaptation. The objective is to modify existing software product while preserving its integrity. This process includes the migration and retirement of the software product. The process ends with the retirement of the software product.

The activities provided in this are specific to the Maintenance Process; however, the process may utilize other processes in this International Standard. If the Development Process (5.3) is utilized, the term developer there is interpreted as maintainer.

The maintainer manages the Maintenance Process at the project level following the Management Process (7.1), which is instantiated in this process; establishes an infrastructure under the process following the Infrastructure Process (7.2); tailors the process for the project following the Tailoring Process (annex A); and manages the process at the organizational level following the Improvement Process (7.3) and the Training Process (7.4). When the maintainer is the supplier of the maintenance service, the maintainer performs the Supply Process (5.2).

List of activities. This process consists of the following activities:

- 1) Process implementation;
- 2) Problem and modification analysis;
- 3) Modification implementation;
- 4) Maintenance review/acceptance;
- 5) Migration;
- 6) Software retirement.

5.5.1 Process implementation. This activity consists of the following tasks:

5.5.1.1 The maintainer shall develop, document, and execute plans and procedures for conducting the activities and tasks of the Maintenance Process.

5.5.1.2 The maintainer shall establish procedures for receiving, recording and tracking problem reports and modification requests from the users and providing feedback to the users. Whenever problems are encountered, they shall be recorded and entered into the Problem Resolution Process (6.8).

5.5.1.3 The maintainer shall implement (or establish organizational interface with) the Configuration Management Process (6.2) for managing modifications to the existing system.

5.5.2 Problem and modification analysis. This activity consists of the following tasks:

5.5.2.1 The maintainer shall analyze the problem report or modification request for its impact on the organization, the existing system, and the interfacing systems for the following:

- a) Type; for example, corrective, improvement, preventive, or adaptive to new environment;
- b) Scope; for example, size of modification, cost involved, time to modify;
- c) Criticality; for example, impact on performance, safety, or security.

5.5.2.2 The maintainer shall replicate or verify the problem.

5.5.2.3 Based upon the analysis, the maintainer shall develop options for implementing the modification.

5.5.2.4 The maintainer shall document the problem/modification request, the analysis results, and implementation options.

5.5.2.5 The maintainer shall obtain approval for the selected modification option as specified in the contract.

5.5.3 Modification implementation. This activity consists of the following tasks:

5.5.3.1 The maintainer shall conduct analysis and determine which documentation, software units, and versions thereof need to be modified. These shall be documented.

5.5.3.2 The maintainer shall enter the Development Process (5.3) to implement the modifications. The requirements of the Development Process shall be supplemented as follows:

- a) Test and evaluation criteria for testing and evaluating the modified and the un-modified parts (software units, components, and configuration items) of the system shall be defined and documented.
- b) The complete and correct implementation of the new and modified requirements shall be ensured. It also shall be ensured that the original, unmodified requirements were not affected. The test results shall be documented.

5.5.4 Maintenance review/acceptance. This activity consists of the following tasks:

5.5.4.1 The maintainer shall conduct review(s) with the organization authorizing the modification to determine the integrity of the modified system.

5.5.4.2 The maintainer shall obtain approval for the satisfactory completion of the modification as specified in the contract.

5.5.5 Migration. This activity consists of the following tasks:

5.5.5.1 If a system or software product (including data) is migrated from an old to a new operational environment, it shall be ensured that any software product or data produced or modified during migration are in accordance with this International Standard.

5.5.5.2 A migration plan shall be developed, documented, and executed. The planning activities shall include users. Items included in the plan shall include the following:

- a) Requirements analysis and definition of migration;
- b) Development of migration tools;
- c) Conversion of software product and data;
- d) Migration execution;
- e) Migration verification;
- f) Support for the old environment in the future.

5.5.5.3 Users shall be given notification of the migration plans and activities. Notifications shall include the following:

- a) Statement of why the old environment is no longer to be supported;
- b) Description of the new environment with its date of availability;
- c) Description of other support options available, if any, once support for the old environment has been removed.

5.5.5.4 Parallel operations of the old and new environments may be conducted for smooth transition to the new environment. During this period, necessary training shall be provided as specified in the contract.

5.5.5.5 When the scheduled migration arrives, notification shall be sent to all concerned. All associated old environment's documentation, logs, and code should be placed in archives.

5.5.5.6 A post-operation review shall be performed to assess the impact of changing to the new environment. The results of the review shall be sent to the appropriate authorities for information, guidance, and action.

5.5.5.7 Data used by or associated with the old environment shall be accessible in accordance with the contract requirements for data protection and audit applicable to the data.

5.5.6 Software retirement. This activity consists of the following tasks:

NOTE – The software product will be retired on the request of the owner.

5.5.6.1 A retirement plan to remove active support by the operation and maintenance organizations shall be developed and documented. The planning activities shall include users. The plan shall address the items listed below. The plan shall be executed.

- a) Cessation of full or partial support after a certain period of time;
- b) Archiving of the software product and its associated documentation;
- c) Responsibility for any future residual support issues;
- d) Transition to the new software product, if applicable;
- e) Accessibility of archive copies of data.

5.5.6.2 Users shall be given notification of the retirement plans and activities. Notifications shall include the following:

- a) Description of the replacement or upgrade with its date of availability;
- b) Statement of why the software product is no longer to be supported;
- c) Description of other support options available, once support has been removed.

5.5.6.3 Parallel operations of the retiring and the new software product should be conducted for smooth transition to the new system. During this period, user training shall be provided as specified in the contract.

5.5.6.4 When the scheduled retirement arrives, notification shall be sent to all concerned. All associated development documentation, logs, and code should be placed in archives, when appropriate.

5.5.6.5 Data used or associated by the retired software product shall be accessible in accordance with the contract requirements for data protection and audit applicable to the data.

6 Supporting life cycle processes

This clause defines the following supporting life cycle processes:

- 1) Documentation process;
- 2) Configuration management process;
- 3) Quality assurance process;
- 4) Verification process;
- 5) Validation process;
- 6) Joint review process;
- 7) Audit process;
- 8) Problem resolution process.

The activities and tasks in a supporting process are the responsibility of the organization performing that process. This organization ensures that the process is in existence and functional.

The organization employing and performing a supporting process manages it at the project level following the Management Process (7.1); establishes an infrastructure under it following the Infrastructure Process (7.2); tailors it for the project following the Tailoring Process (annex A); and manages it at the organizational level following the Improvement Process (7.3) and the Training Process (7.4). Joint Reviews, Audits, Verification, and Validation may be used as techniques of Quality Assurance.

6.1 Documentation process

The Documentation Process is a process for recording information produced by a life cycle process or activity. The process contains the set of activities, which plan, design, develop, produce, edit, distribute, and maintain those documents needed by all concerned such as managers, engineers, and users of the system or software product.

List of activities. This process consists of the following activities:

- 1) Process implementation;
- 2) Design and development;
- 3) Production;
- 4) Maintenance.

6.1.1 Process implementation. This activity consists of the following task:

6.1.1.1 A plan, identifying the documents to be produced during the life cycle of the software product, shall be developed, documented, and implemented. For each identified document, the following shall be addressed:

- a) Title or Name;
- b) Purpose;
- c) Intended audience;
- d) Procedures and responsibilities for inputs, development, review, modification, approval, production, storage, distribution, maintenance, and configuration management;
- e) Schedule for intermediate and final versions.

6.1.2 Design and development. This activity consists of the following tasks:

6.1.2.1 Each identified document shall be designed in accordance with applicable documentation standards for format, content description, page numbering, figure/table placement, proprietary/security marking, packaging, and other presentation items.

6.1.2.2 The source and appropriateness of input data for the documents shall be confirmed. Automated documentation tools may be used.

6.1.2.3 The prepared documents shall be reviewed and edited for format, technical content, and presentation style against their documentation standards. They shall be approved for adequacy by authorized personnel prior to issue.

6.1.3 Production. This activity consists of the following tasks:

6.1.3.1 The documents shall be produced and provided in accordance with the plan. Production and distribution of documents may use paper, electronic, or other media. Master materials shall be stored in accordance with requirements for record retention, security, maintenance, and backup.

6.1.3.2 Controls shall be established in accordance with the Configuration Management Process (6.2).

6.1.4 Maintenance. This activity consists of the following task:

6.1.4.1 The tasks, that are required to be performed when documentation is to be modified, shall be performed (see 5.5). For those documents that are under configuration management, modifications shall be managed in accordance with the Configuration Management Process (6.2).

6.2 Configuration management process

The Configuration Management Process is a process of applying administrative and technical procedures throughout the software life cycle to: identify, define, and baseline software items in a system; control modifications and releases of the items; record and report the status of the items and modification requests; ensure the completeness, consistency, and correctness of the items; and control storage, handling, and delivery of the items.

NOTE – When this process is employed on other software products or entities, the term "software item" below is interpreted accordingly.

List of activities. This process consists of the following activities:

- 1) Process implementation;
- 2) Configuration identification;
- 3) Configuration control;
- 4) Configuration status accounting;
- 5) Configuration evaluation;
- 6) Release management and delivery.

6.2.1 Process implementation. This activity consists of the following task:

6.2.1.1 A configuration management plan shall be developed. The plan shall describe: the configuration management activities; procedures and schedule for performing these activities; the organization(s) responsible for performing these activities; and their relationship with other organizations, such as software development or maintenance. The plan shall be documented and implemented.

NOTE – The plan may be a part of the system configuration management plan.

6.2.2 Configuration identification. This activity consists of the following task:

6.2.2.1 A scheme shall be established for identification of software items and their versions to be controlled for the project. For each software item and its versions, the following shall be identified: the documentation that establishes the baseline; the version references; and other identification details.

6.2.3 Configuration control. This activity consists of the following task:

6.2.3.1 The following shall be performed: identification and recording of change requests; analysis and evaluation of the changes; approval or disapproval of the request; and implementation, verification, and release of the modified software item. An audit trail shall exist, whereby each modification, the reason for the modification, and authorization of the modification can be traced. Control and audit of all accesses to the controlled software items that handle safety or security critical functions shall be performed.

6.2.4 Configuration status accounting. This activity consists of the following task:

6.2.4.1 Management records and status reports that show the status and history of controlled software items including baseline shall be prepared. Status reports should include the number of changes for a project, latest software item versions, release identifiers, the number of releases, and comparisons of releases.

6.2.5 Configuration evaluation. This activity consists of the following task:

6.2.5.1 The following shall be determined and ensured: the functional completeness of the software items against their requirements and the physical completeness of the software items (whether their design and code reflect an up-to-date technical description).

6.2.6 Release management and delivery. This activity consists of the following task:

6.2.6.1 The release and delivery of software products and documentation shall be formally controlled. Master copies of code and documentation shall be maintained for the life of the software product. The code and documentation that contain safety or security critical functions shall be handled, stored, packaged, and delivered in accordance with the policies of the organizations involved.

6.3 Quality assurance process

The Quality Assurance Process is a process for providing adequate assurance that the software products and processes in the project life cycle conform to their specified requirements and adhere to their established plans. To be unbiased, quality assurance needs to have organizational freedom and authority from persons directly responsible for developing the software product or executing the process in the project. Quality assurance may be internal or external depending on whether evidence of product or process quality is demonstrated to the management of the supplier or the acquirer. Quality assurance may make use of the results of other supporting processes, such as Verification, Validation, Joint Reviews, Audits, and Problem Resolution.

List of activities. This process consists of the following activities:

- 1) Process implementation;
- 2) Product assurance;
- 3) Process assurance;
- 4) Assurance of quality systems.

6.3.1 Process implementation. This activity consists of the following tasks:

6.3.1.1 A quality assurance process tailored to the project shall be established. The objectives of the quality assurance process shall be to assure that the software products and the processes employed for providing those software products comply with their established requirements and adhere to their established plans.

6.3.1.2 The quality assurance process should be coordinated with the related Verification (6.4), Validation (6.5), Joint Review (6.6), and Audit (6.7) Processes.

6.3.1.3 A plan for conducting the quality assurance process activities and tasks shall be developed, documented, implemented, and maintained for the life of the contract. The plan shall include the following:

- a) Quality standards, methodologies, procedures, and tools for performing the quality assurance activities (or their references in organization's official documentation);
- b) Procedures for contract review and coordination thereof;
- c) Procedures for identification, collection, filing, maintenance, and disposition of quality records;
- d) Resources, schedule, and responsibilities for conducting the quality assurance activities;
- e) Selected activities and tasks from supporting processes, such as Verification (6.4), Validation (6.5), Joint Review (6.6), Audit (6.7), and Problem Resolution (6.8).

6.3.1.4 Scheduled and on-going quality assurance activities and tasks shall be executed. When problems or non-conformances with contract requirements are detected, they shall be documented and serve as input to the Problem Resolution Process (6.8). Records of these activities and tasks, their execution, problems, and problem resolutions shall be prepared and maintained.

6.3.1.5 Records of quality assurance activities and tasks shall be made available to the acquirer as specified in the contract.

6.3.1.6 It shall be assured that persons responsible for assuring compliance with the contract requirements have the organizational freedom, resources, and authority to permit objective evaluations and to initiate, effect, resolve, and verify problem resolutions.

6.3.2 Product assurance. This activity consists of the following tasks:

6.3.2.1 It shall be assured that all the plans required by the contract are documented, comply with the contract, are mutually consistent, and are being executed as required.

6.3.2.2 It shall be assured that software products and related documentation comply with the contract and adhere to the plans.

6.3.2.3 In preparation for the delivery of the software products, it shall be assured that they have fully satisfied their contractual requirements and are acceptable to the acquirer.

6.3.3 Process assurance. This activity consists of the following tasks:

6.3.3.1 It shall be assured that those software life cycle processes (supply, development, operation, maintenance, and supporting processes including quality assurance) employed for the project comply with the contract and adhere to the plans.

6.3.3.2 It shall be assured that the internal software engineering practices, development environment, test environment, and libraries comply with the contract.

6.3.3.3 It shall be assured that applicable prime-contract requirements are passed down to the subcontractor, and that the subcontractor's software products satisfy prime-contract requirements.

6.3.3.4 It shall be assured that the acquirer and other parties are provided the required support and cooperation in accordance with the contract, negotiations, and plans.

6.3.3.5 It should be assured that software product and process measurements are in accordance with established standards and procedures.

6.3.3.6 It shall be assured that the staff assigned have the skill and knowledge needed to meet the requirements of the project and receive any necessary training.

6.3.4 Assurance of quality systems. This activity consists of the following task:

6.3.4.1 Additional quality management activities shall be assured in accordance with the clauses of ISO 9001 as specified in the contract.

6.4 Verification process

The Verification Process is a process for determining whether the software products of an activity fulfill the requirements or conditions imposed on them in the previous activities. For cost and performance effectiveness, verification should be integrated, as early as possible, with the process (such as supply, development, operation, or maintenance) that employs it. This process may include analysis, review and test.

This process may be executed with varying degrees of independence. The degree of independence may range from the same person or different person in the same organization to a person in a different organization with varying degrees of separation. In the case where the process is executed by an organization independent of the supplier, developer, operator, or maintainer, it is called Independent Verification Process.

List of activities. This process consists of the following activities:

- 1) Process implementation;
- 2) Verification.

6.4.1 Process implementation. This activity consists of the following tasks:

6.4.1.1 A determination shall be made if the project warrants a verification effort and the degree of organizational independence of that effort needed. The project requirements shall be analyzed for criticality. Criticality may be gauged in terms of:

- a) The potential of an undetected error in a system or software requirement for causing death or personal injury, mission failure, or financial or catastrophic equipment loss or damage;
- b) The maturity of and risks associated with the software technology to be used;
- c) Availability of funds and resources.

6.4.1.2 If the project warrants a verification effort, a verification process shall be established to verify the software product.

6.4.1.3 If the project warrants an independent verification effort, a qualified organization responsible for conducting the verification shall be selected. This organization shall be assured of the independence and authority to perform the verification activities.

6.4.1.4 Based upon the scope, magnitude, complexity, and criticality analysis above, target life cycle activities and software products requiring verification shall be determined. Verification activities and tasks defined in 6.4.2, including associated methods, techniques, and tools for performing the tasks, shall be selected for the target life cycle activities and software products.

6.4.1.5 Based upon the verification tasks as determined, a verification plan shall be developed and documented. The plan shall address the life cycle activities and software products subject to verification, the required verification tasks for each life cycle activity and software product, and related resources, responsibilities, and schedule. The plan shall address procedures for forwarding verification reports to the acquirer and other involved organizations.

6.4.1.6 The verification plan shall be implemented. Problems and non-conformances detected by the verification effort shall be entered into the Problem Resolution Process (6.8). All problems and non-conformances shall be resolved. Results of the verification activities shall be made available to the acquirer and other involved organizations.

6.4.2 Verification. This activity consists of the following tasks:

6.4.2.1 Contract verification. The contract shall be verified considering the criteria listed below:

- a) The supplier has the capability to satisfy the requirements.
- b) The requirements are consistent and cover user needs.
- c) Adequate procedures for handling changes to requirements and escalating problems are stipulated.
- d) Procedures and their extent for interface and cooperation among the parties are stipulated, including ownership, warranty, copyright and confidentiality.
- e) Acceptance criteria and procedures are stipulated in accordance with requirements.

NOTE – This activity may be used in the contract review (see 6.3.1.3 b).

6.4.2.2 Process verification. The process shall be verified considering the criteria listed below:

- a) Project planning requirements are adequate and timely.
- b) Processes selected for the project are adequate, implemented, being executed as planned, and compliant with the contract.
- c) The standards, procedures, and environments for the project's processes are adequate.
- d) The project is staffed and personnel trained as required by the contract.

6.4.2.3 Requirements verification. The requirements shall be verified considering the criteria listed below:

- a) The system requirements are consistent, feasible, and testable.
- b) The system requirements have been appropriately allocated to hardware items, software items, and manual operations according to design criteria.
- c) The software requirements are consistent, feasible, testable, and accurately reflect system requirements.
- d) The software requirements related to safety, security, and criticality are correct as shown by suitably rigorous methods.

6.4.2.4 Design verification. The design shall be verified considering the criteria listed below:

- a) The design is correct and consistent with and traceable to requirements.
- b) The design implements proper sequence of events, inputs, outputs, interfaces, logic flow, allocation of timing and sizing budgets, and error definition, isolation, and recovery.
- c) Selected design can be derived from requirements.
- d) The design implements safety, security, and other critical requirements correctly as shown by suitably rigorous methods.

6.4.2.5 Code verification. The code shall be verified considering the criteria listed below:

- a) The code is traceable to design and requirements, testable, correct, and compliant with requirements and coding standards.

- b) The code implements proper event sequence, consistent interfaces, correct data and control flow, completeness, appropriate allocation timing and sizing budgets, and error definition, isolation, and recovery.
- c) Selected code can be derived from design or requirements.
- d) The code implements safety, security, and other critical requirements correctly as shown by suitably rigorous methods.

6.4.2.6 Integration verification. The integration shall be verified considering the criteria listed below:

- a) The software components and units of each software item have been completely and correctly integrated into the software item.
- b) The hardware items, software items, and manual operations of the system have been completely and correctly integrated into the system.
- c) The integration tasks have been performed in accordance with an integration plan.

6.4.2.7 Documentation verification. The documentation shall be verified considering the criteria listed below:

- a) The documentation is adequate, complete, and consistent.
- b) Documentation preparation is timely.
- c) Configuration management of documents follows specified procedures.

6.5 Validation process

The Validation Process is a process for determining whether the requirements and the final, as-built system or software product fulfills its specific intended use. Validation may be conducted in earlier stages. This process may be conducted as a part of Software Acceptance Support (5.3.13).

This process may be executed with varying degrees of independence. The degree of independence may range from the same person or different person in the same organization to a person in a different organization with varying degrees of separation. In the case where the process is executed by an organization independent of the supplier, developer, operator, or maintainer, it is called Independent Validation Process.

List of activities. This process consists of the following activities:

- 1) Process implementation;
- 2) Validation.

6.5.1 Process implementation. This activity consists of the following tasks:

6.5.1.1 A determination shall be made if the project warrants a validation effort and the degree of organizational independence of that effort needed.

6.5.1.2 If the project warrants a validation effort, a validation process shall be established to validate the system or software product. Validation tasks defined below, including associated methods, techniques, and tools for performing the tasks, shall be selected.

6.5.1.3 If the project warrants an independent effort, a qualified organization responsible for conducting the effort shall be selected. The conductor shall be assured of the independence and authority to perform the validation tasks.

6.5.1.4 A validation plan shall be developed and documented. The plan shall include, but is not limited to, the following:

- a) Items subject to validation;
- b) Validation tasks to be performed;
- c) Resources, responsibilities, and schedule for validation;
- d) Procedures for forwarding validation reports to the acquirer and other parties.

6.5.1.5 The validation plan shall be implemented. Problems and non-conformances detected by the validation effort shall be entered into the Problem Resolution Process (6.8). All problems and non-conformances shall be resolved. Results of the validation activities shall be made available to the acquirer and other involved organizations.

6.5.2 Validation. This activity shall consist of the following tasks:

6.5.2.1 Prepare selected test requirements, test cases, and test specifications for analyzing test results.

6.5.2.2 Ensure that these test requirements, test cases, and test specifications reflect the particular requirements for the specific intended use.

6.5.2.3 Conduct the tests in subclauses 6.5.2.1 and 6.5.2.2, including:

- a) Testing with stress, boundary, and singular inputs;
- b) Testing the software product for its ability to isolate and minimize the effect of errors; that is, graceful degradation upon failure, request for operator assistance upon stress, boundary, and singular conditions;

- c) Testing that representative users can successfully achieve their intended tasks using the software product.

6.5.2.4 Validate that the software product satisfies its intended use.

6.5.2.5 Test the software product as appropriate in selected areas of the target environment.

6.6 Joint review process

The Joint Review Process is a process for evaluating the status and products of an activity of a project as appropriate. Joint reviews are at both project management and technical levels and are held throughout the life of the contract. This process may be employed by any two parties, where one party (reviewing party) reviews another party (reviewed party).

List of activities: This process consists of the following activities:

- 1) Process implementation;
- 2) Project management reviews;
- 3) Technical reviews.

6.6.1 Process implementation. This activity consists of the following tasks:

6.6.1.1 Periodic reviews shall be held at predetermined milestones as specified in the project plan(s). *Ad hoc* reviews should be called when deemed necessary by either party.

6.6.1.2 All resources required to conduct the reviews shall be agreed on by the parties. These resources include personnel, location, facilities, hardware, software, and tools.

6.6.1.3 The parties should agree on the following items at each review: meeting agenda, software products (results of an activity) and problems to be reviewed; scope and procedures; and entry and exit criteria for the review.

6.6.1.4 Problems detected during the reviews shall be recorded and entered into the Problem Resolution Process (6.8) as required.

6.6.1.5 The review results shall be documented and distributed. The reviewing party will acknowledge to the reviewed party the adequacy (for example, approval, disapproval, or contingent approval) of the review results.

6.6.1.6 The parties shall agree on the outcome of the review and any action item responsibilities and closure criteria.

6.6.2 Project management reviews. This activity consists of the following task:

6.6.2.1 Project status shall be evaluated relative to the applicable project plans, schedules, standards, and guidelines. The outcome of the review should be discussed between the two parties and should provide for the following:

- a) Making activities progress according to plan, based on an evaluation of the activity or software product status;
- b) Maintaining global control of the project through adequate allocation of resources;
- c) Changing project direction or determining the need for alternate planning;
- d) Evaluating and managing the risk issues that may jeopardize the success of the project.

6.6.3 Technical reviews. This activity consists of the following task:

6.6.3.1 Technical reviews shall be held to evaluate the software products or services under consideration and provide evidence that:

- a) They are complete.
- b) They comply with their standards and specifications.

- c) Changes to them are properly implemented and affect only those areas identified by the Configuration Management Process (6.2).
- d) They are adhering to applicable schedules.
- e) They are ready for the next activity.
- f) The development, operation, or maintenance is being conducted according to the plans, schedules, standards, and guidelines of the project.

.....

6.7 Audit process

The Audit Process is a process for determining compliance with the requirements, plans, and contract as appropriate. This process may be employed by any two parties, where one party (auditing party) audits the software products or activities of another party (audited party).

List of activities. This process consists of the following activities:

- 1) Process implementation;
- 2) Audit.

6.7.1 Process implementation. This activity consists of the following tasks:

6.7.1.1 Audits shall be held at predetermined milestones as specified in the project plan(s).

6.7.1.2 Auditing personnel shall not have any direct responsibility for the software products and activities they audit.

6.7.1.3 All resources required to conduct the audits shall be agreed by the parties. These resources include supporting personnel, location, facilities, hardware, software, and tools.

6.7.1.4 The parties should agree on the following items at each audit: agenda; software products (and results of an activity) to be reviewed; audit scope and procedures; and entry and exit criteria for the audit.

6.7.1.5 Problems detected during the audits shall be recorded and entered into the Problem Resolution Process (6.8) as required.

6.7.1.6 After completing an audit, the audit results shall be documented and provided to the audited party. The audited party shall acknowledge to the auditing party any problems found in the audit and related problem resolutions planned.

6.7.1.7 The parties shall agree on the outcome of the audit and any action item responsibilities and closure criteria.

6.7.2 Audit. This activity consists of the following task:

6.7.2.1 Audits shall be conducted to ensure that:

- a) As-coded software products (such as a software item) reflect the design documentation.
- b) The acceptance review and testing requirements prescribed by the documentation are adequate for the acceptance of the software products.
- c) Test data comply with the specification.
- d) Software products were successfully tested and meet their specifications.
- e) Test reports are correct and discrepancies between actual and expected results have been resolved.
- f) User documentation complies with standards as specified.
- g) Activities have been conducted according to applicable requirements, plans, and contract.
- h) The costs and schedules adhere to the established plans.

6.8 Problem resolution process

The Problem Resolution Process is a process for analyzing and resolving the problems (including non-conformances), whatever their nature or source, that are discovered during the execution of development, operation, maintenance, or other processes. The objective is to provide a timely, responsible, and documented means to ensure that all discovered problems are analyzed and resolved and trends are recognized.

List of activities. This process consists of the following activities:

- 1) Process implementation;
- 2) Problem resolution.

6.8.1 Process implementation. This activity consists of the following task:

6.8.1.1 A problem resolution process shall be established for handling all problems (including non-conformances) detected in the software products and activities. The process shall comply with the following requirements:

- a) The process shall be closed-loop, ensuring that: all detected problems are promptly reported and entered into the Problem Resolution Process; action is initiated on them; relevant parties are advised of the existence of the problem as appropriate; causes are identified, analyzed, and, where possible, eliminated; resolution and disposition are achieved; status is tracked and reported; and records of the problems are maintained as stipulated in the contract.
- b) The process should contain a scheme for categorizing and prioritizing the problems. Each problem should be classified by the category and priority to facilitate trend analysis and problem resolution.
- c) Analysis shall be performed to detect trends in the problems reported.
- d) Problem resolutions and dispositions shall be evaluated: to evaluate that problems have been resolved, adverse trends have been reversed, and changes have been correctly implemented in the appropriate software products and activities; and to determine whether additional problems have been introduced.

6.8.2 Problem resolution. This activity consists of the following task:

6.8.2.1 When problems (including non-conformances) have been detected in a software product or an activity, a problem report shall be prepared to describe each problem detected. The problem report shall be used as part of the closed-loop process described above: from detection of the problem, through investigation, analysis and resolution of the problem and its cause, and onto trend detection across problems.

7 Organizational life cycle processes

This clause defines the following organizational life cycle processes:

- 1) Management process;
- 2) Infrastructure process;
- 3) Improvement process;
- 4) Training process.

The activities and tasks in an organizational process are the responsibility of the organization using that process. The organization ensures that the process is in existence and functional.

7.1 Management process

The Management Process contains the generic activities and tasks, which may be employed by any party that has to manage its respective process(es). The manager is responsible for product management, project management, and task management of the applicable process(es), such as the acquisition, supply, development, operation, maintenance, or supporting process.

List of activities: This process consists of the following activities:

- 1) Initiation and scope definition;
- 2) Planning;
- 3) Execution and control;
- 4) Review and evaluation;
- 5) Closure.

7.1.1 Initiation and scope definition. This activity consists of the following tasks:

7.1.1.1 The management process shall be initiated by establishing the requirements of the process to be undertaken.

7.1.1.2 Once the requirements are established, the manager shall establish the feasibility of the process by checking that the resources (personnel, materials, technology, and environment) required to execute and manage the process are available, adequate, and appropriate and that the time-scales to completion are achievable.

7.1.1.3 As necessary, and by agreement of all parties concerned, the requirements of the process may be modified at this point to achieve the completion criteria.

7.1.2 Planning. This activity consists of the following task:

7.1.2.1 The manager shall prepare the plans for execution of the process. The plans associated with the execution of the process shall contain descriptions of the associated activities and tasks and identification of the software products that will be provided. These plans shall include, but are not limited to, the following:

- a) Schedules for the timely completion of tasks;
- b) Estimation of effort;
- c) Adequate resources needed to execute the tasks;
- d) Allocation of tasks;
- e) Assignment of responsibilities;
- f) Quantification of risks associated with the tasks or the process itself;
- g) Quality control measures to be employed throughout the process;
- h) Costs associated with the process execution;
- i) Provision of environment and infrastructure.

7.1.3 Execution and control. This activity consists of the following tasks:

7.1.3.1 The manager shall initiate the implementation of the plan to satisfy the objectives and criteria set, exercising control over the process.

7.1.3.2 The manager shall monitor the execution of the process, providing both internal reporting of the process progress and external reporting to the acquirer as defined in the contract.

7.1.3.3 The manager shall investigate, analyze, and resolve the problems discovered during the execution of the process. The resolution of problems may result in changes to plans. It is the manager's responsibility to ensure the impact of any changes is determined, controlled, and monitored. Problems and their resolution shall be documented.

7.1.3.4 The manager shall report, at agreed points, the progress of the process, declaring adherence to the plans and resolving instances of the lack of progress. These include internal and external reporting as required by the organizational procedures and the contract.

7.1.4 Review and evaluation. This activity consists of the following tasks:

7.1.4.1 The manager shall ensure that the software products and plans are evaluated for satisfaction of requirements.

7.1.4.2 The manager shall assess the evaluation results of the software products, activities, and tasks completed during the execution of the process for achievement of the objectives and completion of the plans.

7.1.5 Closure. This activity consists of the following tasks:

7.1.5.1 When all software products, activities, and tasks are completed, the manager shall determine whether the process is complete taking into account the criteria as specified in the contract or as part of organization's procedure.

7.1.5.2 The manager shall check the results and records of the software products, activities, and tasks employed for completeness. These results and records shall be archived in a suitable environment as specified in the contract.

7.2 Infrastructure process

The Infrastructure Process is a process to establish and maintain the infrastructure needed for any other process. The infrastructure may include hardware, software, tools, techniques, standards, and facilities for development, operation, or maintenance.

List of activities. This process consists of the following activities:

- 1) Process implementation;
- 2) Establishment of the infrastructure;
- 3) Maintenance of the infrastructure.

7.2.1 Process implementation. This activity consists of the following tasks:

7.2.1.1 The infrastructure should be defined and documented to meet the requirements of the process employing this process, considering the applicable procedures, standards, tools, and techniques.

7.2.1.2 The establishment of the infrastructure should be planned and documented.

7.2.2 Establishment of the infrastructure. This activity consists of the following tasks:

7.2.2.1 The configuration of the infrastructure should be planned and documented. Functionality, performance, safety, security, availability, space requirements, equipment, costs, and time constraints should be considered.

7.2.2.2 The infrastructure shall be installed in time for execution of the relevant process.

7.2.3 Maintenance of the infrastructure. This activity consists of the following task:

7.2.3.1 The infrastructure shall be maintained, monitored, and modified as necessary to ensure that it continues to satisfy the requirements of the process employing this process. As part of maintaining the infrastructure, the extent to which the infrastructure is under configuration management shall be defined.

7.3 Improvement process

The Improvement Process is a process for establishing, assessing, measuring, controlling, and improving a software life cycle process.

List of activities. This process consists of the following activities:

- 1) Process establishment;
- 2) Process assessment;
- 3) Process improvement.

7.3.1 Process establishment. This activity consists of the following task:

7.3.1.1 The organization shall establish a suite of organizational processes for all software life cycle processes as they apply to its business activities. The processes and their application to specific cases shall be documented in organization's publications. As appropriate, a process control mechanism should be established to develop, monitor, control, and improve the process(es).

7.3.2 Process assessment. This activity consists of the following tasks:

7.3.2.1 A process assessment procedure should be developed, documented, and applied. Assessment records should be kept and maintained.

7.3.2.2 The organization shall plan and carry out reviews of the processes at appropriate intervals to ensure their continuing suitability and effectiveness in the light of assessment results.

7.3.3 Process improvement. This activity consists of the following tasks:

7.3.3.1 The organization shall effect such improvements to its processes as it determines to be necessary as a result of process assessment and review. Process documentation should be updated to reflect improvement in the organizational processes.

7.3.3.2 Historical, technical, and evaluation data should be collected and analyzed to gain an understanding of the strengths and weaknesses of the employed processes. These analyses should be used as feedback to improve these processes, to recommend changes in the direction of the projects (or subsequent projects), and to determine technology advancement needs.

7.3.3.3 Quality cost data should be collected, maintained, and used to improve the organization's processes as a management activity. These data shall serve the purpose of establishing the cost of both the prevention and resolution of problems and non-conformity in software products and services.

7.4 Training process

The Training Process is a process for providing and maintaining trained personnel. The acquisition, supply, development, operation, or maintenance of software products is largely dependent upon knowledgeable and skilled personnel. For example: developer personnel should have essential training in software management and software engineering. It is, therefore, imperative that personnel training be planned and implemented early so that trained personnel are available as the software product is acquired, supplied, developed, operated, or maintained.

List of activities. This process consists of the following activities:

- 1) Process implementation;
- 2) Training material development;
- 3) Training plan implementation.

7.4.1 Process implementation. This activity consists of the following task:

7.4.1.1 A review of the project requirements shall be conducted to establish and make timely provision for acquiring or developing the resources and skills required by the management and technical staff. The types and levels of training and categories of personnel needing training shall be determined. A training plan, addressing implementation schedules, resource requirements, and training needs, should be developed and documented.

7.4.2 Training material development. This activity consists of the following task:

7.4.2.1 Training manuals, including presentation materials used in providing training, should be developed.

7.4.3 Training plan implementation. This activity consists of the following tasks:

7.4.3.1 The training plan shall be implemented to provide training to personnel. Training records should be maintained.

7.4.3.2 It should be ensured that the right mix and categories of appropriately trained personnel are available for the planned activities and tasks in a timely manner.

Annex A
(normative)
Tailoring process

The Tailoring Process is a process for performing basic tailoring of this International Standard for a software project. This annex provides requirements for tailoring this International Standard.

List of activities. This process consists of the following activities:

- 1) Identifying project environment;
- 2) Soliciting inputs;
- 3) Selecting processes, activities, and tasks;
- 4) Documenting tailoring decisions and rationale.

A.1 Identifying project environment. This activity consists of the following task:

A.1.1 Characteristics of the project environment that are going to influence tailoring shall be identified. Some of the characteristics may be: life cycle model; current system life cycle activity; system and software requirements; organizational policies, procedures and strategies; size, criticality and types of the system, software product or service; and number of personnel and parties involved.

A.2 Soliciting inputs. This activity consists of the following task:

A.2.1 Inputs from the organizations that are to be affected by the tailoring decisions shall be solicited. Users, support personnel, contracting officers, potential bidders should be involved in tailoring.

A.3 Selecting processes, activities, and tasks. This activity consists of the following tasks:

A.3.1 The processes, activities, and tasks that are to be performed shall be decided. These include the documentation to be developed and who are to be responsible for them. For this purpose, this International Standard should be evaluated against relevant data gathered in clauses A.1 and A.2.

A.3.2 The processes, activities, and tasks that were decided upon in A.3.1 but not provided in this International Standard shall be specified in the contract itself. Organizational life cycle processes (clause 7) should be evaluated to determine whether they could provide for these processes, activities, and tasks.

A.3.3 In this International Standard, requirements are indicated by tasks that contain "shall" or "will." These tasks should be carefully considered for whether they should be kept or deleted for a given project or a given business sector. Factors to be considered include but are not limited to: risk, cost, schedule, performance, size, criticality, and human interface.

A.4 Documenting tailoring decisions and rationale. This activity consists of the following task:

A.4.1 All tailoring decisions shall be documented together with the rationale for the decisions.

Annex B (informative) Guidance on tailoring

No two projects are the same. Variations in organizational policies and procedures, acquisition methods and strategies, project size and complexity, system requirements and development methods, among other things, influence how a system is acquired, developed, operated, or maintained. This International Standard is written for a general project to accommodate such variations as much as possible. Therefore, in the interest of cost reduction and quality improvement, this International Standard should be tailored for an individual project. All parties involved in the project should be involved in tailoring.

B.1 General tailoring guidance. This clause provides guidance on tailoring this International Standard and is not exhaustive. This clause may be used to perform first-level tailoring of this International Standard for a given business area; for example, aviation, nuclear, medical, military, country, or organization. The second-level tailoring should be performed for each specific project or contract.

B.2 Tailoring of the Development Process

The Development Process (5.3) needs special attention, because this process may be used by different parties with different objectives. As a first-level tailoring of this process, the following is recommended:

- a) For a software product that is embedded in or integral to the system: all the activities in the process should be considered; and it should be clarified whether the developer is required to *perform or support* the system activities.
- b) For a stand-alone software product, the system activities (5.3.2, 5.3.3, 5.3.10 and 5.3.11) may not be required but should be considered.

B.3 Tailoring of the evaluation-related activities

Persons who are involved in any activity of the life cycle of a project or a process, conduct evaluations either on their own or other's software products and activities. This International Standard groups these evaluations into five categories, which are listed below. The first four evaluation categories are at project level; the last one is at organizational level. These evaluations should be selected and tailored proportional to the scope, magnitude, complexity, and criticality of the project or of the organization. The problem, non-conformance, and improvement reports from these evaluations feed into the Problem Resolution Process (6.8).

- a) Process-internal evaluations (evaluation tasks in 5.1 to 5.5). These are conducted by personnel performing the assigned tasks within the process during their day-to-day activities.
- b) Verification (6.4) and Validation (6.5). Conducted by the acquirer, the supplier, or an independent party, to verify and validate the products in varying depth depending on the project. These evaluations do not duplicate or replace other evaluations, but supplement them.
- c) Joint Reviews (6.6) and Audits (6.7). These are conducted in a joint forum by the reviewing and reviewed parties to evaluate status and compliance of products and activities on a pre-agreed to schedule.
- d) Quality Assurance (6.3). Conducted by personnel independent of the personnel directly responsible for developing the software product or executing the process. The goal is to *independently assure* conformance of the software products and processes with the contract requirements and adherence to the established plans. This process may use the results from a, b, and c above as inputs. This process may coordinate its activities with those of a, b, and c.
- e) Improvement (7.3). Conducted by an organization for efficient management and self-improvement of its process. This is conducted regardless of project or contract requirements.

B.4 Tailoring and application considerations

The paragraphs in this clause outline broad tailoring and application considerations for key project characteristics. Neither the considerations nor the characteristics are exhaustive and represent only current thinking. Figure B.1 provides an example of the application of this International Standard.

Organizational policies. Determine which organizational policies are relevant and applicable, such as on computer languages, safety and security, hardware reserve requirements, and risk management. The clauses of this International Standard related to these organizational policies should be kept.

Acquisition strategy. Determine which acquisition strategies are relevant and applicable for the project, such as types of contract, more than one contractor, involvement of subcontractors and verification and validation agents, level of acquirer's involvement with contractors, and evaluation of contractors' capabilities. The clauses of this International Standard related to these strategies should be kept.

Support concept. Determine which support concepts are relevant and applicable, such as expected length of support, degree of change, and whether the acquirer or the supplier will support. If the software product will have a long support life or is expected to change significantly, all documentation requirements should be considered. It is advisable to have the documentation automated.

Life Cycle model(s). Determine which life cycle model(s) are relevant and applicable for the project, such as Waterfall, evolutionary, builds, pre-planned product improvement, Spiral. All such models prescribe certain processes and activities that may be performed sequentially, repeated, and combined; in these models, the life cycle activities in this International Standard should be mapped to the selected model(s). For evolutionary, build, and pre-planned product improvement models, the outputs of one project activity feed into the next. In these cases, the documentation should be complete at the end of an activity or a task.

Parties involved. Determine or identify which parties are involved in the project, such as acquirer, supplier, developer, subcontractor, verification agent, and validation agent, maintainer; and the number of personnel. All the requirements related to organizational interfaces between two parties are under consideration; for example, acquirer to developer and supplier to verification or validation agent. A large project involving many (tens or hundreds) persons needs significant management oversight and control. Tools such as internal and independent evaluations, reviews, audits and inspections, and data collection are important for a large project. For small projects, these controls may be excessive.

System life cycle activity. Determine which current system life cycle activities are relevant and applicable, such as acquirer's project initiation, supplier's development, and maintenance. Some scenarios:

Acquirer is initiating or defining system requirements. Feasibility studies and prototyping of requirements and design may be conducted. Software code for prototypes may be developed, which may or may not be used later in the development of software products performed under contract. System requirements and preliminary software requirements may be developed. In these cases, the Development Process (5.3) may be used as a guidance rather than requirement; the rigor of qualification and evaluation may not be needed; and joint reviews and audits may not be needed.

Developer is producing software products under contract. In this case all Development Process (5.3) requirements should be considered during tailoring.

Maintainer is modifying software products. The Maintenance Process (5.5) is under consideration. Parts of the Development Process (5.3) may be used as mini-processes.

System-Level characteristics. Determine which system-level characteristics are relevant and applicable, such as number of subsystems and configuration items. If the system has many subsystems or configuration items, the Development Process (5.3) should be carefully tailored for each subsystem and configuration item. All interface and integration requirements should be considered.

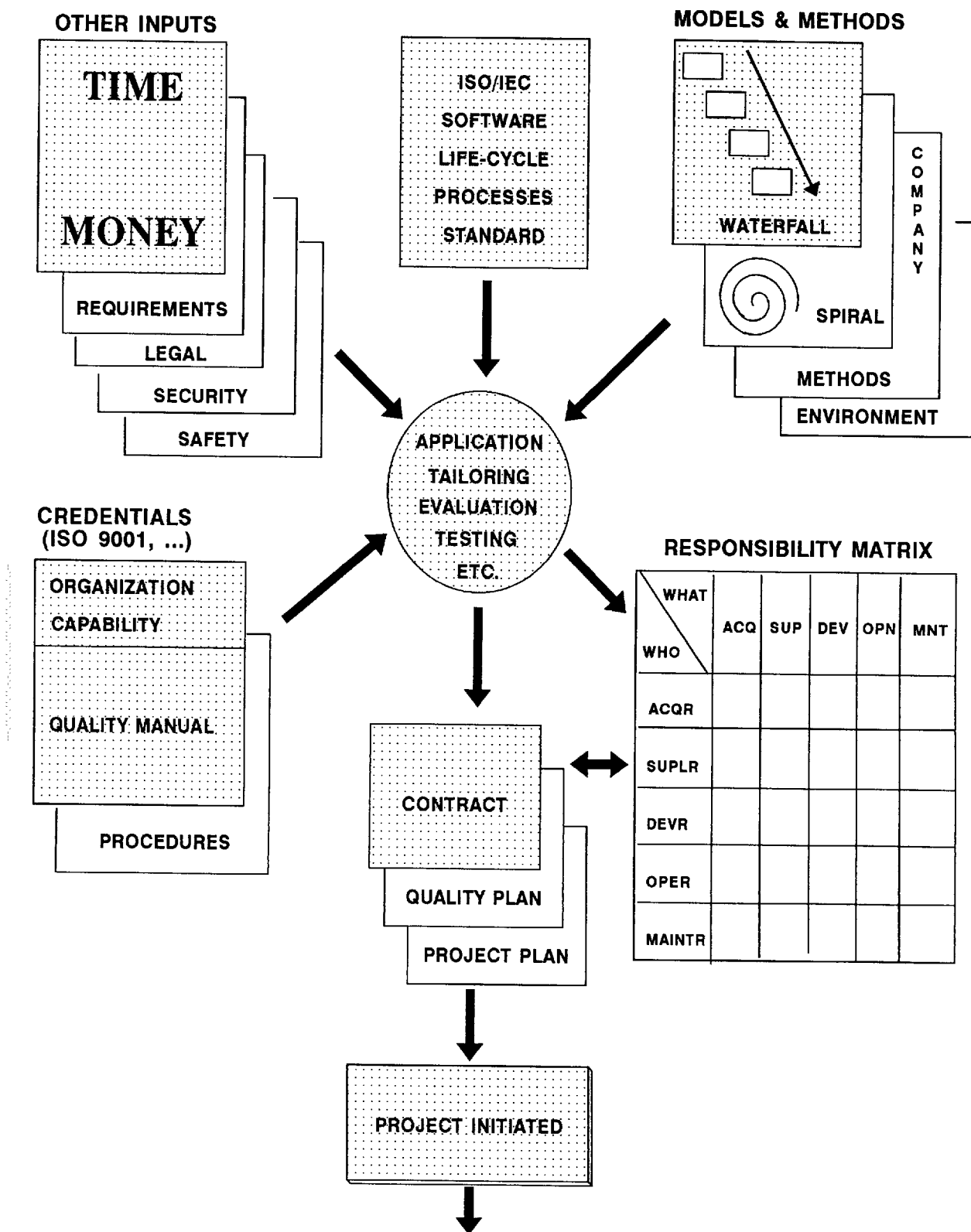


Figure B.1 An Example of Application of the International Standard

Software-Level characteristics. Determine which software-level characteristics are relevant and applicable, such as number of software items, types, size and criticality of software products, and technical risks. If the software product has many software items, components, and units, the Development Process (5.3) should be carefully tailored for each software item. All interface and integration requirements should be considered.

Determine which types of software product are involved, as different types of software product may require different tailoring decisions. Some examples:

- a) New development. All of the requirements, particularly the Development Process (5.3), should be under consideration.
- b) Use of off-the-shelf software product "as is." The full Development Process (5.3) may be excessive. Performance, documentation, proprietary, usage, ownership, warranty and licensing rights, and future support related to the software product should be evaluated.
- c) Modification of off-the-shelf software product. Documentation may not be available. Depending on the criticality and expected future changes, the Development Process (5.3) should be used via the Maintenance Process (5.5). Performance, documentation, proprietary, ownership, usage, warranty and licensing rights, and future support related to the software product should be evaluated.
- d) Software or firmware product embedded in or integral to a system. Since such a software product is a part of a larger system, the system-related activities in the Development Process (5.3) should be considered. In the system-related activities, only one verb "perform" or "support" needs to be selected. If the software or firmware product is not likely to be modified in the future, extent of documentation needs should be carefully examined.
- e) Software product that is stand-alone. Since such a software product is not a part of a system, the system-related activities in the Development Process (5.3) need not be considered. Documentation needs, particularly for maintenance, should be carefully examined.
- f) Non-deliverable software product. As no items are being acquired, supplied, or developed, no provision of this International Standard except 5.3.1.5 of the Development Process (5.3) should be considered. However, if the acquirer decides to acquire a piece of such a software product for future operation and maintenance, then this software product should be treated as in b or c above.

Other considerations.

The more dependent the system is upon the software product operating correctly and being finished on time, the more management control should be imposed via testing, reviews, audits, verification, validation, and so on. Conversely, much management control of non-critical or small software product may not be cost-effective.

Development of software product may have technical risks. If the software technology used is not mature, software product being developed is unprecedented or complex, or software product contains safety, security, or other critical requirements, then rigorous specification, design, testing, and evaluations may be needed. Independent verification and validation may be important.

Annex C (informative) Guidance on processes and organizations

This annex, to promote understandability, presents a discussion on the processes, organizations, and their relationships under key viewpoints.

C.1 Processes under key points of view

This International Standard contains the processes that are applicable throughout the life cycle of software. However, these processes may be used in different ways by different organizations and parties with different views and objectives. This clause presents the processes and their relationships under key points of view. See 4.1.1 for synopses of the processes.

Figure C.1 depicts the software life cycle processes and their relationships under different views of the usage of this International Standard. The basic views shown are: contract, management, operating, engineering, and supporting. Under the contract view, acquirer and supplier parties negotiate and enter into a contract and employ the Acquisition Process and Supply Process respectively. Under the management view, the acquirer, supplier, developer, operator, maintainer, or other party manages its respective process. Under the operating view, the operator provides software operation service for the users. Under the engineering view, the developer or maintainer conducts its respective engineering tasks to produce or modify software products. Under the supporting view, parties (such as configuration management, quality assurance) provide supporting services to others in fulfilling specific, unique tasks. Also shown (see the bottom box) are the organizational processes; these are employed by an organization at the corporate level to establish and implement an underlying structure made up of associated life cycle process(es) and personnel and continuously improve them.

Figure C.2 presents the primary (top, left box), supporting (top, right box), and organizational (bottom box) life cycle processes and their constituent activity names under different views. A numeral prefixed to a process refers to the section number in this International Standard.

The contract view has two life cycle processes (see the upper shaded box under the Primary Life Cycle Processes): an Acquisition Process for the acquirer and a Supply Process for the supplier. Each process shows its constituent activities. These processes define the tasks for the acquirer and the supplier respectively from the contractual viewpoint.

The engineering view has two life cycle processes (see the lower, left-bottom shaded box in the Primary Life Cycle Processes): a Development Process and a Maintenance Process. Each process shows its constituent activities. The Development Process is employed by development engineers for producing software products. The Maintenance Process is employed by maintenance engineers for modifying the software and keeping it current.

The operating view has one life cycle process (see the lower, right shaded box in the Primary Life Cycle Processes): an Operation Process and its constituent activities. The Operation Process is employed for operating the software for its users.

The quality management view has six life cycle processes (see the shaded box in Supporting Life Cycle Processes): Quality Assurance Process; Verification Process; Validation Process; Joint Review Process; and Audit Process. Their constituent activities are not shown. These quality related processes are employed for managing quality throughout the software life cycle. The Verification; Validation; Joint Review; and Audit processes may be employed by different parties separately and as techniques of the Quality Assurance Process as well.

The management view has one process (see the shaded box in Organizational Life Cycle Processes): a Management Process that is used by any organization for managing its respective process. Its constituent activities are shown.

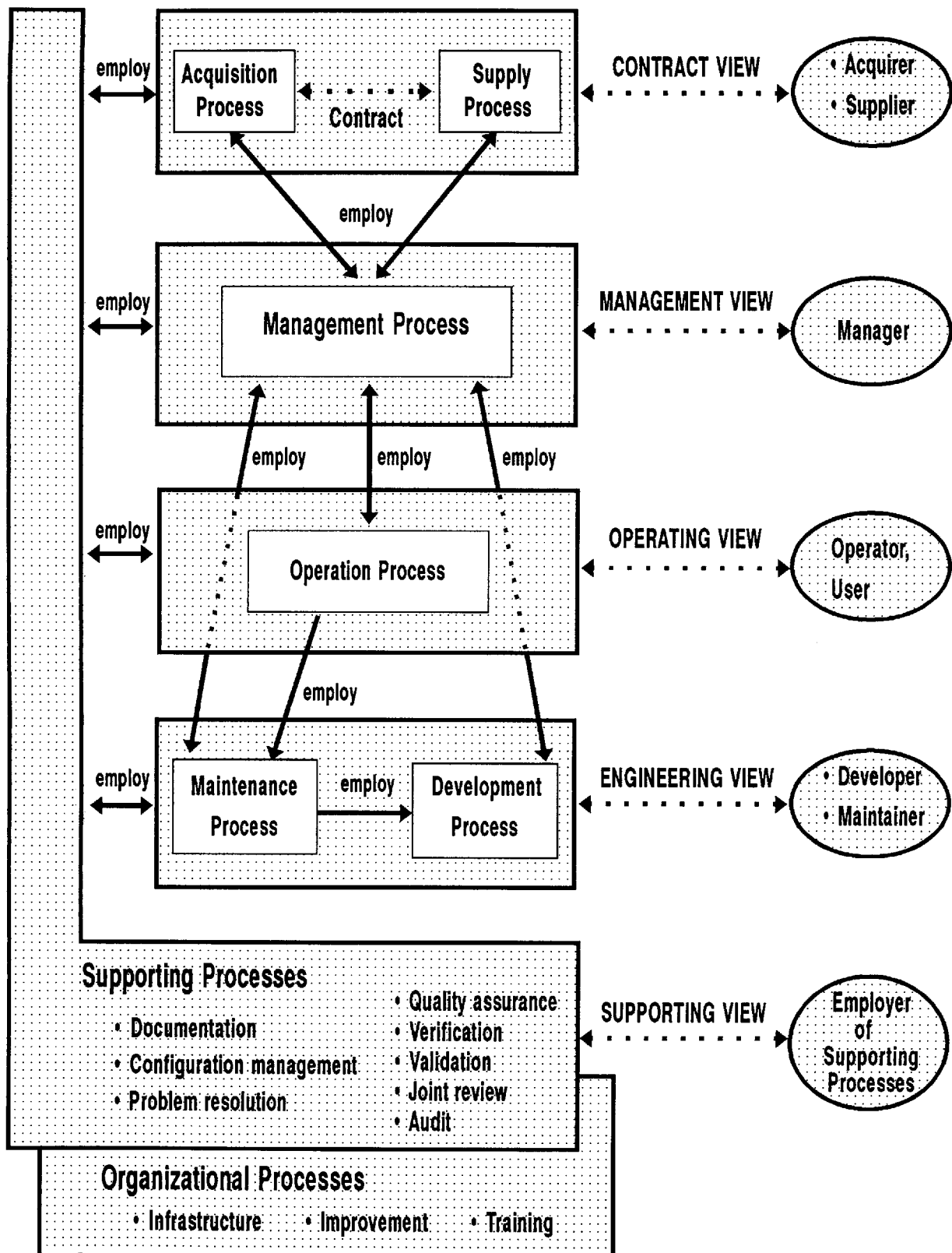
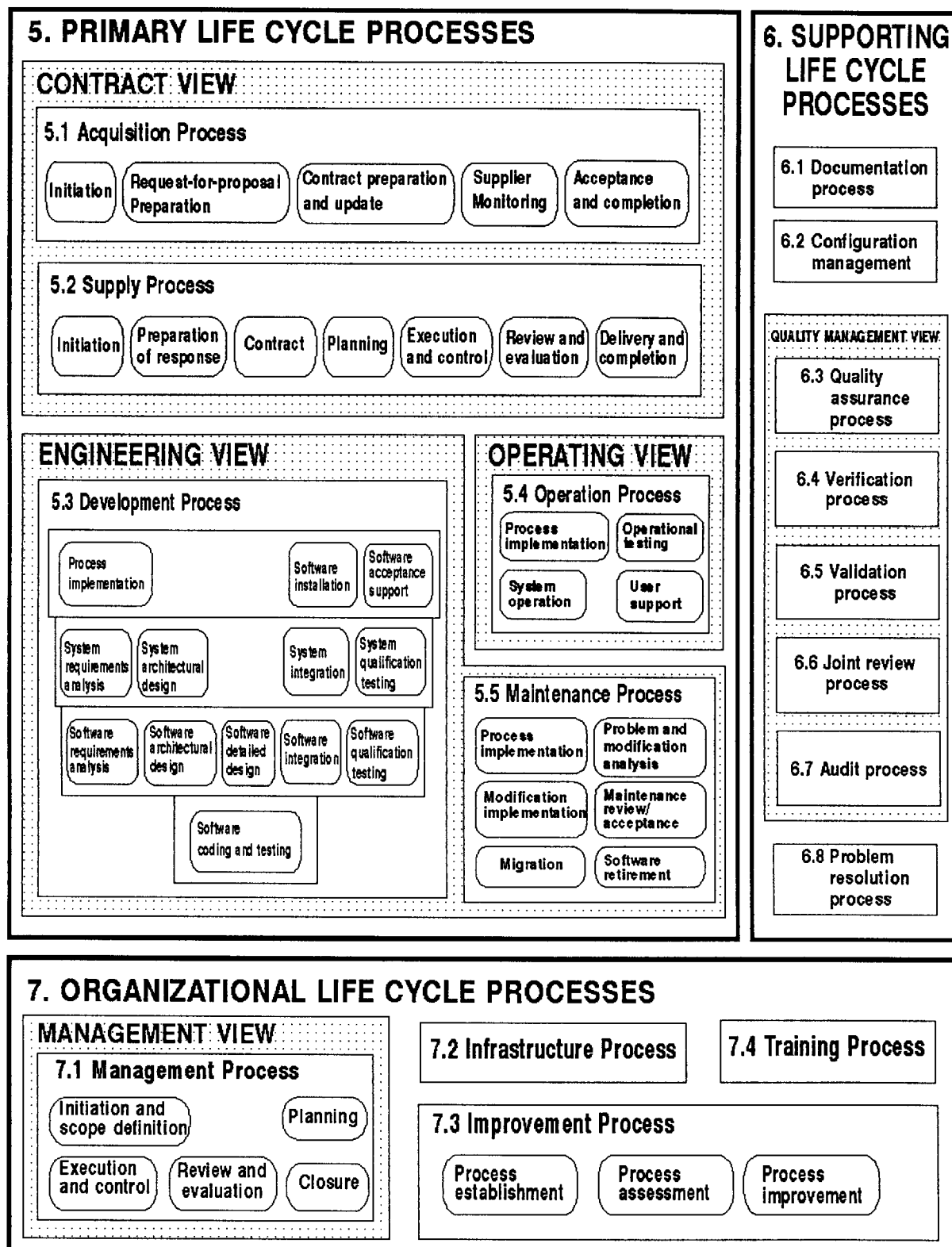


Figure C.1 Software Life-Cycle Processes -- Roles and Relationships



The position order of activities does not mean time order.
Names of activities in the Development Process are not names of development phases.

Figure C.2 Software Life Cycle Processes, Views and Activities

C.2 Processes, organizations, and relationships

The processes and organizations (or parties) are only related functionally. They do not dictate a structure for an organization (or a party).

In this International Standard, the terms "organization" and "party" are *nearly* synonymous. An organization is a body of persons organized for some specific purpose, as a club, union, corporation, or society. When an organization, as a whole or a part, enters into a contract, it is a party. Organizations are separate bodies, but the parties may be from the same organization or from separate organizations.

An organization or a party gets its name from the process it performs; for example, it is called an acquirer when it performs the Acquisition Process.

An organization may perform one process or more than one process; a process may be performed by one organization or more than one organization. *Under one contract or application of this International Standard*, a given party should not perform both the Acquisition Process and the Supply Process, but it can perform other processes.

In this International Standard itself, the relationships between the processes are only static. The more important dynamic, real-life relationships between the processes, between the parties, and between the processes and the parties are automatically established when this International Standard is applied on software projects. Each process (and the party performing it) contributes to the software project in its own unique way. The Acquisition Process (and the acquirer) contributes by defining the system, which would contain software product. The Supply Process (and the supplier) contributes by providing the software product or service on which that system would depend. The Development Process (and the developer) contributes by "looking" to the system for correct derivation and definition of software product, by supporting proper integration of the software product back into the system, and by developing the software product in between. The Operation Process (and the operator) contributes by operating the software product in the system's environment for the benefit of the users, the business, and the mission. The Maintenance Process (and the maintainer) contributes by maintaining and sustaining the software product for operational fitness and by providing support and advice to the user community. Each supporting or organizational process contributes by providing unique, specialized functions to other processes as needed.

Annex D
(informative)
Bibliography

ISO/IEC 12119: 1994, *Information technology - Software packages - Quality requirements and testing.*

ICS 35.080

Descriptors: data processing, data processing equipment, computers, computer software, life cycle.

Price based on 57 pages
